



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - K141502

PENCARIAN INITIAL BASIC FEASIBLE SOLUTION PADA TRANSPORTATION PROBLEM DENGAN MENGUNAKAN ALGORITMA INCESSANT ALLOCATION METHOD MODIFICATION

MOHAMMAD RASYID KAROMI
NRP 5111440000101

Dosen Pembimbing
Bilqis Amaliah, S.Kom., M.Kom.
Victor Hariadi, S.Si, M.Kom.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - K141502

PENCARIAN INITIAL BASIC FESIBLE SOLUTION PADA TRANSPORTATION PROBLEM DENGAN MENGUNAKAN ALGORITMA INCESSANT ALLOCATION METHOD MODIFICATION

MOHAMMAD RASYID KAROMI
NRP 5111440000101

Dosen Pembimbing
Bilqis Amaliah, S.Kom., M.Kom.
Victor Hariadi, S.Si, M.Kom.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - K141502

FIND THE INITIAL BASIC FEASIBLE SOLUTION OF TRANSPORTATION PROBLEM WITH INCESSANT ALLOCATION METHOD MODIFICATION ALGORITHM

MOHAMMAD RASYID KAROMI
NRP 5111440000101

Advisor
Bilqis Amaliah, S.Kom., M.Kom.
Victor Hariadi, S.Si, M.Kom.

INFORMATICS DEPARTMENT
Faculty of Information and Communication Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

**PENCARIAN INITIAL BASIC FEASIBLE SOLUTION
PADA TRANSPORTATION PROBLEM DENGAN
MENGUNAKAN ALGORITMA INCESSANT
ALLOCATION METHOD MODIFICATION**

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Dasar dan Terapan Komputasi
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

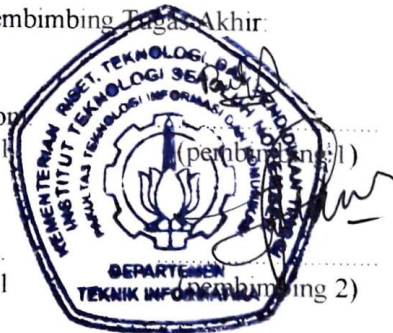
MOHAMMAD RASYID KAROMI

NRP: 5111440000101

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Bilqis Amaliah, S.Kom., M.Kom.
NIP: 19751914 200112 2 001

Victor Hariadi, S.Si., M.Kom.
NIP: 19691228 199412 1 001



**SURABAYA
JULI 2018**

[Halaman ini sengaja dikosongkan]

PENCARIAN INITIAL BASIC FEASIBLE SOLUTION PADA TRANSPORTATION PROBLEM DENGAN MENGUNAKAN ALGORITMA INCESSANT ALLOCATION METHOD MODIFICATION

Nama Mahasiswa : Mohammad Rasyid Karomi
NRP : 5111440000101
Departemen : Informatika FTIK-ITS
Dosen Pembimbing 1 : Bilqis Amaliah, S.Kom., M.Kom.
Dosen Pembimbing 2 : Victor Hariadi, S.Si, M.Kom.

ABSTRAK

Pada masa kini, dunia perindustrian membutuhkan sistem pendistribusian produk yang tidak memakan biaya besar. Hal ini memicu para peneliti untuk melakukan riset dalam menciptakan algoritma yang efektif untuk mencari biaya terkecil dalam sistem pendistribusian produk. Para peneliti kemudian menamakan permasalahan matematika linier ini sebagai Transportation Problem. Dalam Transportation Problem, tujuan utamanya adalah menciptakan algoritma yang dapat menghasilkan biaya terkecil dalam mengirimkan sejumlah kuantitas barang yang mulanya tersimpan di beberapa tempat pemasok ke tempat-tempat tujuan. Terdapat beberapa faktor yang mempengaruhi besar-kecilnya biaya yaitu banyaknya pemasok, banyaknya kebutuhan pada tempat tujuan pengiriman, dan estimasi biaya tiap rute.

Salah satu penelitian Transportation Problem adalah algoritma Incessant Allocation Method (IAM). IAM merupakan algoritma yang mengutamakan rute dengan biaya terkecil secara berurutan hingga semua pasokan terkirim atau semua kebutuhan terpenuhi. Pada penelitian ini, algoritma IAM akan dimodifikasi pada bagian inisiasi perulangan dan bagian pemilihan

prioritasnya. Algoritma IAM yang baru dinamakan algoritma Incessant Allocation Method - Initiation Looping (IAM-IL) dan Incessant Allocation Method Priority Value (IAM-PV). Algoritma-algoritma tersebut diimplementasikan dalam bentuk program berbahasa C dengan format input sejumlah supply, sejumlah demand, dan cost tiap rute. Output yang diharapkan dari program ini adalah nilai alokasi tiap rute dan total biaya pengalokasian tersebut.

Algoritma IAM dan IAM-PV diujicobakan terhadap 40 studi kasus dari beberapa paper lain. Pada hasil uji penelitian ini, persentase akurasi algoritma IAM sebesar 27,5%, sedangkan persentase akurasi algoritma IAM-PV naik 32,5% menjadi sebesar 60%. Selain itu, didapatkan juga persentase deviasi algoritma IAM sebesar 72,5%, sedangkan nilai persentase deviasi algoritma IAM-PV turun 11,889% menjadi sebesar 2,699%. Sehingga pada penelitian ini dapat disimpulkan bahwa algoritma IAM-PV lebih mendekati hasil optimal daripada algoritma IAM

Kata kunci: algoritma incesant allocation method, algoritma incesant allocation method priority value, sistem pendistribusian produk, transportation problem

FIND THE INITIAL BASIC FEASIBLE SOLUTION OF TRANSPORTATION PROBLEM WITH INCESSANT ALLOCATION METHOD MODIFICATION ALGORITHM

Name : Mohammad Rasyid Karomi
NRP : 5114 100 101
Department : Informatics FTIK-ITS
Supervisor I : Bilqis Amaliah, S.Kom., M.Kom.
Supervisor II : Victor Hariadi, S.Si, M.Kom.

ABSTRACT

In the present, the world of industry requires a system of distribution of products that do not cost big. This prompted researchers to research in creating an effective algorithm to find the smallest cost in the product distribution system. The researchers then named this linear mathematical problem as a Transportation Problem. In Transportation Problem, the main objective is to create algorithms that can generate the smallest cost in delivering quantities of goods originally stored in multiple supplier places to the destination places. There are several factors that affect the size of the cost of the number of suppliers, the number of needs at the destination of delivery, and the estimated cost of each route.

One of the research of Transportation Problem is the Incessant Allocation Method (IAM) algorithm. IAM is an algorithm that prioritizes the least costly route in sequence until all supplies are delivered or all needs are met. In this study, the IAM algorithm will be modified on the part of the loop initiation and the priority selection section. The new IAM algorithm is called the Incessant Allocation Method-Initiation Looping (IAM-IL) algorithm and the Incessant Allocation Method Priority Value (IAM-PV). The algorithms are implemented in the form of C-language programs with input format of supply, demand, and cost

of each route. The expected output of this program is the allocation value of each route and the total cost of the allocation.

IAM and IAM-PV algorithms were tested against 40 case studies from several other papers. In the results of this study, the percentage accuracy of IAM algorithm is 27.5%, while the percentage accuracy of IAM-PV algorithm up 32.5% to 60%. In addition, IAM algorithm percentage deviation percentage is 72.5%, while the percentage deviation of IAM-PV algorithm is decreased 11,889% to 2,699%. So in this study it can be concluded that IAM-PV algorithm is closer to optimal results than IAM algorithm

Keywords: incesant allocation method algorithm, incesant allocation method priority value algorithm, product distribution system, transportation problem

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT karena atas karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

PENCARIAN INITIAL BASIC FEASIBLE SOLUTION PADA TRANSPORTATION PROBLEM DENGAN MENGUNAKAN ALGORITMA INCESSANT ALLOCATION METHOD MODIFICATION

Melalui lembar ini, penulis ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Bapak, Ibu, Kakak, sekeluarga tercinta yang selalu memberikan doa serta dukungan kepada penulis untuk menyelesaikan tugas akhir ini.
2. Ibu Bilqis Amaliah, S.Kom., M.Kom. selaku dosen pembimbing tugas akhir pertama yang telah membimbing dan memberi banyak masukan dalam pengerjaan tugas akhir ini.
3. Bapak Victor Hariadi, S.Si, M.Kom. selaku dosen pembimbing tugas akhir kedua yang telah memberikan masukan serta koreksi dalam pengerjaan tugas akhir.
4. Bapak Radityo Anggoro S.Kom, M.Sc. selaku mahasiswa jurusan Informatika ITS yang telah memberikan dukungan selama perkuliahan
5. Ibu Adhatus Solichah S.Kom, M.Sc selaku dosen wali yang senantiasa memberikan dukungan baik selama perkuliahan maupun selama pengerjaan tugas akhir ini.
6. Bapak dan Ibu dosen serta seluruh civitas Departemen Informatika yang telah memberikan pelajaran dan pengalaman selama menjadi mahasiswa di Departemen Informatika.

7. Kawan-kawan TC14, C1E, serta seluruh anggota HMTC yang sudah menemani, mendukung dan memberikan pelajaran dalam organisasi maupun perkuliahan.
8. Teman-teman serumah di Jojoran. Faiq, Kevin, Akhyar, Hamka, Naufan, Hari, Hakim, Anandi, Haidar, Trastian, Bayu, Aldi, Dimas, Rian, dan yang lainnya untuk semangat, dukungan serta rasa kekeluargaan yang ada.
9. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu per satu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyelesaikan tugas akhir ini. Namun, penulis mohon maaf apabila terdapat kekurangan ataupun kesalahan yang penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan untuk ke depannya.

Surabaya, Juli 2018

Mohammad Rasyid Karomi

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER	xxv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Permasalahan.....	2
1.3 Batasan Permasalahan	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Metodologi	3
1.7 Sistematika Penulisan.....	5
BAB II DASAR TEORI.....	7
2.1 <i>Transportation Problem</i>	7
2.2 <i>Initial Base Feasible Solution (IBFS)</i>	9
2.3 Algoritma Northwest Corner Method	10
2.4 Algoritma Least Cost Method	12
2.5 Algoritma Vogel Approximation Method	14
2.6 Algoritma Incessant Allocation Method	16

BAB III ANALISIS DAN PERANCANGAN SISTEM.....	23
3.1 Analisis Algoritma IAM.....	23
3.2 Perubahan <i>Looping</i> pada Inisiasi.....	25
3.3 Perbandingan Prioritas pada <i>Cell</i>	31
BAB IV IMPLEMENTASI.....	39
4.1 Lingkungan Implementasi.....	39
4.2 Implementasi Algoritma IAM-PV	39
4.2.1 Implementasi Tahap Menghitung Priority Value	40
4.2.2 Implementasi Tahap Memilih Priority Value	40
4.2.3 Implementasi Tahap Alokasi Supply Demand	
Habis	41
4.2.4 Implementasi Tahap Alokasi Supply Habis	42
4.2.5 Implementasi Tahap Alokasi Demand Habis	44
4.2.6 Implementasi Tahap Menghitung Total Cost	46
BAB V PENGUJIAN DAN EVALUASI	47
5.1 Lingkungan Pengujian.....	47
5.2 Data Uji Coba.....	47
5.3 Hasil Uji Coba	47
1. Uji Coba Soal 1	47
2. Uji Coba Soal 2	48
3. Uji Coba Soal 3	49
4. Uji Coba Soal 4	49
5. Uji Coba Soal 5	50
6. Uji Coba Soal 6	50
7. Uji Coba Soal 7	51

8.	Uji Coba Soal 8	51
9.	Uji Coba Soal 9	52
10.	Uji Coba Soal 10	52
11.	Uji Coba Soal 11	53
12.	Uji Coba Soal 12	53
13.	Uji Coba Soal 13	54
14.	Uji Coba Soal 14	54
15.	Uji Coba Soal 15	55
16.	Uji Coba Soal 16	55
17.	Uji Coba Soal 17	56
18.	Uji Coba Soal 18	56
19.	Uji Coba Soal 19	57
20.	Uji Coba Soal 20	57
21.	Uji Coba Soal 21	58
22.	Uji Coba Soal 22	58
23.	Uji Coba Soal 23	59
24.	Uji Coba Soal 24	59
25.	Uji Coba Soal 25	60
26.	Uji Coba Soal 26	60
27.	Uji Coba Soal 27	61
28.	Uji Coba Soal 28	61
29.	Uji Coba Soal 29	62
30.	Uji Coba Soal 30	62
31.	Uji Coba Soal 31	63

32.	Uji Coba Soal 32	63
33.	Uji Coba Soal 33	64
34.	Uji Coba Soal 34	64
35.	Uji Coba Soal 35	65
36.	Uji Coba Soal 36	65
37.	Uji Coba Soal 37	66
38.	Uji Coba Soal 38	66
39.	Uji Coba Soal 39	67
40.	Uji Coba Soal 40	67
41.	Uji Coba Soal Khusus	68
5.4	Evaluasi Algoritma IAM-PV	68
BAB VI KESIMPULAN DAN SARAN.....		77
6.1.	Kesimpulan.....	77
6.2.	Saran.....	77
DAFTAR PUSTAKA.....		79
LAMPIRAN		81
BIODATA PENULIS.....		127

DAFTAR GAMBAR

Gambar 1 Diagram Model Transportation Problem.....	7
Gambar 2 <i>Flowchart</i> algoritma NWCM.	11
Gambar 3 <i>Flowchart</i> algoritma LCM.	13
Gambar 4 <i>Flowchart</i> algoritma VAM.....	15
Gambar 5 <i>Flowchart</i> algoritma IAM.	16
Gambar 6 <i>Flowchart</i> algoritma IAM - IL.	26
Gambar 7 <i>Flowchart</i> algoritma IAM - PV.....	35
Gambar 8 Grafik	35

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 1 Matrix Transportation Problem	8
Tabel 2 Matrix Transportation Problem pada Problem Set 1	18
Tabel 3 Tabel solusi algortima IAM tahap inisiasi	20
Tabel 4 Tabel solusi algortima IAM tahap perulangan I.....	20
Tabel 5 Tabel solusi algortima IAM tahap perulangan II	21
Tabel 6 Tabel solusi algortima IAM tahap perulangan III	21
Tabel 7 Tabel solusi algortima IAM tahap perulangan IV	22
Tabel 8 Tabel solusi algortima IAM tahap perulangan V	22
Tabel 9 Tabel solusi optimal Problem Set-1	23
Tabel 10 Tabel solusi algortima IAM tahap perulangan II	24
Tabel 11 Tabel solusi algortima IAM-IL tahap inisiasi	28
Tabel 12 Tabel solusi algortima IAM-IL tahap perulangan I....	29
Tabel 13 Tabel solusi algortima IAM-IL tahap perulangan II ..	29
Tabel 14 Tabel solusi algortima IAM-IL tahap perulangan III .	30
Tabel 15 Tabel solusi algortima IAM-IL tahap perulangan IV .	30
Tabel 16 Tabel solusi algortima IAM-IL tahap perulangan V ..	31
Tabel 17 Tabel solusi optimal Problem Set-1	32
Tabel 18 Tabel priorirty value tahap inisiasi	36
Tabel 19 Tabel solusi algortima IAM-PV tahap inisiasi	36
Tabel 20 Tabel priorirty value tahap perulangan I	37
Tabel 21 Tabel solusi algortima IAM-PV tahap perulangan I ..	37
Tabel 22 Tabel priorirty value tahap perulangan II.....	38

Tabel 23 Tabel solusi algoritma IAM-PV tahap perulangan II.	38
Tabel 24 Spesifikasi Perangkat	39
Tabel 25 Hasil tiap algoritma pada Problem Set 5.1	48
Tabel 26 Hasil tiap algoritma pada Problem Set 5.2.....	48
Tabel 27 Hasil tiap algoritma pada Problem Set 5.3.....	49
Tabel 28 Hasil tiap algoritma pada Problem Set 5.4	49
Tabel 29 Hasil tiap algoritma pada Problem Set 5.5	50
Tabel 30 Hasil tiap algoritma pada Problem Set 5.6.....	50
Tabel 31 Hasil tiap algoritma pada Problem Set 5.7.....	51
Tabel 32 Hasil tiap algoritma pada Problem Set 5.8.....	51
Tabel 33 Hasil tiap algoritma pada Problem Set 5.9.....	52
Tabel 34 Hasil tiap algoritma pada Problem Set 5.10.....	52
Tabel 35 Hasil tiap algoritma pada Problem Set 5.11	53
Tabel 36 Hasil tiap algoritma pada Problem Set 5.12.....	53
Tabel 37 Hasil tiap algoritma pada Problem Set 5.13.....	54
Tabel 38 Hasil tiap algoritma pada Problem Set 5.14.....	54
Tabel 39 Hasil tiap algoritma pada Problem Set 5.15.....	55
Tabel 40 Hasil tiap algoritma pada Problem Set 5.16.....	55
Tabel 41 Hasil tiap algoritma pada Problem Set 5.17.....	56
Tabel 42 Hasil tiap algoritma pada Problem Set 5.18.....	56
Tabel 43 Hasil tiap algoritma pada Problem Set 5.19	57
Tabel 44 Hasil tiap algoritma pada Problem Set 5.20.....	57
Tabel 45 Hasil tiap algoritma pada Problem Set 5.21	58
Tabel 46 Hasil tiap algoritma pada Problem Set 5.22.....	58

Tabel 47 Hasil tiap algoritma pada Problem Set 5.23	59
Tabel 48 Hasil tiap algoritma pada Problem Set 5.24	59
Tabel 49 Hasil tiap algoritma pada Problem Set 5.25	60
Tabel 50 Hasil tiap algoritma pada Problem Set 5.26	60
Tabel 51 Hasil tiap algoritma pada Problem Set 5.27	61
Tabel 52 Hasil tiap algoritma pada Problem Set 5.28	61
Tabel 53 Hasil tiap algoritma pada Problem Set 5.29	62
Tabel 54 Hasil tiap algoritma pada Problem Set 5.30	62
Tabel 55 Hasil tiap algoritma pada Problem Set 5.31	63
Tabel 56 Hasil tiap algoritma pada Problem Set 5.32	63
Tabel 57 Hasil tiap algoritma pada Problem Set 5.33	64
Tabel 58 Hasil tiap algoritma pada Problem Set 5.34	64
Tabel 59 Hasil tiap algoritma pada Problem Set 5.35	65
Tabel 60 Hasil tiap algoritma pada Problem Set 5.36	65
Tabel 61 Hasil tiap algoritma pada Problem Set 5.37	66
Tabel 62 Hasil tiap algoritma pada Problem Set 5.38	66
Tabel 63 Hasil tiap algoritma pada Problem Set 5.39	67
Tabel 64 Hasil tiap algoritma pada Problem Set 5.40	67
Tabel 65 Hasil tiap algoritma pada Problem Set Khusus	67
Tabel 66 Nilai Akurasi Algoritma	68
Tabel 67 Nilai Persentase Berhasil dan Gagal	69
Tabel 68 Nilai Persentase Deviasi dan Improvement	69

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 1 Implementasi menghitung Priority Value	40
Kode Sumber 2 Implementasi memilih Priority Value	41
Kode Sumber 3 Implementasi alokasi supply demand habis	42
Kode Sumber 4 Implementasi alokasi supply habis	44
Kode Sumber 5 Implementasi alokasi demand habis	46
Kode Sumber 6 Implementasi menghitung Total Cost	46

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Salah satu masalah yang sering terjadi di bidang perindustrian adalah masalah distribusi produk. Dalam melakukan proses distribusi produk, diperlukan perencanaan agar biaya distribusi dapat ditekan menjadi seminimal mungkin [2]. Permasalahan ini mendorong para peneliti untuk melakukan riset dalam mencari *minimal cost*. Permasalahan ini kemudian dikenal sebagai Transportation Problem.

Secara garis besar, terdapat 2 tahapan dalam menyelesaikan Transportation Problem, yaitu inisialisasi dengan cara mencari *Initial Basic Feasible Solution* (IBFS) dan kemudian melakukan optimalisasi dengan metode *Stepping Stone* (SS). IBFS merupakan solusi awal yang diberikan dengan menggunakan algoritma tertentu sedemikian rupa sehingga biaya pengiriman pada Transportation Problem mendekati angka paling minimal [1]. Berbagai riset telah dilakukan oleh peneliti untuk mencari algoritma IBFS, salah satunya adalah Incessant Allocation Method (IAM).

IAM merupakan algoritma IBFS yang menggunakan metode pengalokasian terus menerus pada tiap rute dengan memperhatikan biaya terkecil hingga semua rute terisi. Algoritma IAM sering digunakan karena mudah dipahami dan dapat diaplikasikan dalam pemrograman. Namun, pada paper yang ditulis oleh Mollah Mesbahuddin Ahmed, algoritma IAM masih memiliki prosentase error yang cukup besar yaitu 5% [2]. Dalam tugas akhir ini, algoritma IAM akan dikembangkan dengan cara mengubah inisiasi perulangannya. Diharapkan dengan dikembangkannya algoritma IAM ini dapat menghasilkan biaya yang lebih kecil dan error yang lebih rendah.

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana memodifikasi algoritma *Incessant Allocation Method* (IAM) sehingga dapat menghasilkan biaya yang lebih kecil daripada algoritma IAM yang asli?
2. Berapa besar nilai kenaikan akurasi algoritma IAM yang telah dikembangkan dibandingkan dengan algoritma IAM yang asli?

1.3 Batasan Permasalahan

Batasan masalah pada tugas akhir ini antara lain:

1. *Tools* yang akan digunakan untuk mengecek jawaban optimal adalah ToraSystem 7th Edition *Tools* ini melakukan langkah inisiasi dan optimalisasi (*Stepping Stone*) sekaligus sehingga hasil yang didapatkan pasti optimal.
2. *Tools* yang akan digunakan untuk mengimplementasikan IAM hasil modifikasi ke dalam pemrograman adalah Dev-C++ 5.4.1.
3. Algoritma IBFS yang dibahas pada penelitian ini hanya akan ditelaah pada proses inisialisasi saja.
4. Data soal yang digunakan untuk diuji coba adalah 40 soal, terdiri atas 33 soal dari paper dan 7 soal di-*generate* secara random.
5. Dimensi soal yang dapat diselesaikan oleh program algoritma IAM modifikasi adalah tidak terbatas, namun karena pada dimensi maksimal Tora System adalah 100x100, maka pada penelitian ini dimensi terbesar soal uji coba adalah 100x100
6. Algoritma IBFS lain yang digunakan sebagai pembanding adalah algoritma *North-West*, algoritma *Least-Cost*, algoritma *Vogel*, dan algoritma IAM.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini antara lain:

1. Memodifikasi algoritma *Incessant Allocation Method* (IAM) sehingga dapat menghasilkan biaya yang lebih kecil dibandingkan dengan algoritma IAM yang asli.
2. Menghitung besar nilai kenaikan akurasi algoritma IAM yang telah dikembangkan dibandingkan dengan algoritma IAM yang asli.

1.5 Manfaat

Hasil akhir tugas akhir ini dapat menjadi pembanding dengan algoritma IAM yang asli, sehingga diharapkan dapat menjadi referensi bagi peneliti lain dalam mengembangkan algoritma IBFS. Selain itu, program yang telah dibuat dalam tugas akhir ini dapat digunakan untuk mendapatkan IBFS berbagai soal, sehingga pada studi kasus nyata dapat digunakan untuk menyelesaikan masalah *Transportation Problem*.

1.6 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Studi literatur

Pada studi literatur ini akan dipelajari sejumlah referensi yang relevan terhadap tugas akhir yang akan dikerjakan. Studi literatur ini didapatkan dari buku, internet serta materi-materi kuliah yang berhubungan dengan metode yang akan digunakan. Hal-hal yang akan dipelajari antara lain mengenai *Transportation Problem*, *Initial Base Feasible Soultion* (IBFS) dan algoritma *Incessant Allocation Method* (IAM).

2. Analisis dan desain metode

Metode yang akan digunakan adalah modifikasi dari *Incessant Allocation Method* (IAM). Bagian yang dikembangkan pada algoritma IAM adalah pada proses inisiasi setiap perulangan dan juga pada pemberian bobot prioritas rute yang akan dipilih. Algoritma ini untuk

selanjutnya akan dikembangkan dalam bentuk kode sumber.

3. Implementasi

Pada tugas akhir ini, algoritma IAM modifikasi yang akan dibuat akan diimplementasikan menjadi program berbahasa C++. *Tools* yang akan digunakan adalah Dev C++ 5.4.1 karena telah menyediakan *library* yang dibutuhkan. Selain IAM yang telah dimodifikasi, algoritma IAM sendiri juga akan dibuat programnya, karena pada penelitian IAM peneliti tidak menyediakan *tools* untuk uji coba dan hanya menghitung secara manual.

4. Pengujian dan evaluasi

Uji coba akan dilakukan terhadap 40 soal studi kasus dari paper IAM, 33 soal studi kasus dari beberapa paper lain, dan 7 soal di-*generate* secara random. Algoritma yang akan digunakan sebagai pembanding antara lain *North-West*, *Least-Cost*, *Vogel*, dan *IAM*. Langkah pertama dalam pengujian adalah mencoba algoritma IAM dan IAM modifikasi yang telah dibuat programnya pada keseluruhan 40 studi kasus. Langkah berikutnya adalah menggunakan *tools* ToraSystem 7th Edition untuk mendapatkan hasil dari algoritma *North-West*, *Least-Cost*, *Vogel*. Selain itu, dengan *tools* ToraSystem juga akan didapatkan jawaban optimal setiap studi kasus. Jawaban optimal ini akan berperan sebagai kunci jawaban setiap studi kasus yang nantinya akan menjadi patokan terhadap akurasi masing-masing algoritma.

5. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan proses dokumentasi dan pembuatan laporan dari seluruh konsep, tinjauan pustaka, metode, implementasi, proses yang telah dilakukan, pengujian, evaluasi dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir.

1.7 Sistematika Penulisan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini:

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan tugas akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

Bab II Dasar Teori

Bab ini menjelaskan beberapa teori yang dijadikan penunjang dan berhubungan dengan pokok pembahasan yang mendasari pembuatan tugas akhir.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan sistem yang akan dibangun. Perancangan sistem meliputi perancangan data dan alur proses dari sistem itu sendiri.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan sistem yang telah ditentukan sebelumnya.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dari metode yang ditawarkan dalam tugas akhir untuk mengetahui kesesuaian metode dengan data yang ada.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang telah dilakukan. Bab ini juga membahas

saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

Lampiran

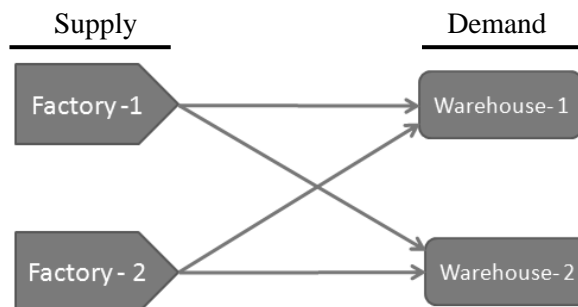
Merupakan bab tambahan yang berisi data atau daftar istilah yang penting pada tugas akhir ini.

BAB II DASAR TEORI

Bab ini membahas teori-teori yang menjadi dasar pembuatan tugas akhir.

2.1 *Transportation Problem*

Transportation Problem adalah salah satu dari masalah Pemrograman Linier di mana tujuannya adalah mengirimkan sejumlah kuantitas barang yang mulanya tersimpan di beberapa tempat pemasok ke tempat-tempat tujuan dengan biaya sekecil mungkin [5]. Dalam *Transportation Problem*, diketahui jumlah barang pasokan (*supply*) dari masing-masing pemasok dan jumlah barang yang harus dikirimkan (*demand*) ke masing-masing tempat tujuan. Selain itu, masing-masing rute dari setiap tempat pemasok menuju setiap tempat tujuan membutuhkan biaya (*cost*) yang berbeda-beda. Contoh *Transportation Problem* telah direpresentasikan dalam bentuk model pada gambar 1.



Gambar 1. Diagram Model Transportation Problem

Gambar 1, merepresentasikan sebuah *Transportation Problem* pada proses penyimpanan produk industri. Produk yang dihasilkan dari beberapa pabrik nantinya akan disimpan di dalam ruang-ruang penyimpanan dengan kapasitas tertentu. Tujuan dari *Transportation Problem* ini adalah untuk

memindahkan semua produk tersebut dari pabrik-pabrik menuju ruang penyimpanan dengan biaya seminimal mungkin. Untuk mencari solusi dalam *Transportation Problem*, langkah pertama yang harus dilakukan adalah mengkonversi *Transportation Problem* menjadi bentuk matrix seperti pada tabel 1 dibawah ini

Tabel 1. Matrix Transportation Problem

Factories	Warehouses				Supply
	W_1	W_2	...	W_n	
F_1	C_{11} A_{11}	C_{12} A_{12}	C_{1n} A_{1n}	S_1
F_2	C_{21} A_{21}	C_{22} A_{22}	C_{2n} A_{2n}	S_2
...
F_m	C_{m1} A_{m1}	C_{m2} A_{m2}	C_{mn} A_{mn}	S_n
Demand	D_1	D_2	...	D_n	

F_i = Factory pada index ke - i

W_i = Warehouse pada index ke - i

S_i = Supply dari Factory pada index ke - i

D_i = Demand dari Warehouse pada index ke - i

C_{ij} = Cost pada rute dari F_i menuju W_j

A_{ij} = Alokasi pada rute dari F_i menuju W_j

Supply direpresentasikan dalam baris, sedangkan *demand* akan direpresentasikan dalam kolom. Perpotongan antara baris dan kolom adalah *cell*. *Cell* mempresentasikan rute dari pemasok ke tujuan. Pada setiap *cell*, terdapat nilai *cost* untuk melewati rute tersebut. Alokasi akan diberikan pada setiap *cell* tersebut. Secara umum, soal *Transportation Problem* dalam notasi matematika disajikan sebagai berikut :

$$\begin{aligned} [C_{ij}] \ m \times n &= [c_{11} \ c_{12} \ \dots \ c_{1n}], [c_{21} \ \dots \ c_{2n}], \dots, [c_{m1} \ c_{m2} \ \dots \ c_{mn}] \\ [S_i] \ m \times 1 &= [s_1, s_2, \dots, s_m] \\ [D_j] \ 1 \times n &= [d_1, d_2, \dots, d_n] \end{aligned}$$

Berikut ini adalah rumus untuk menghitung nilai *cost* total setelah alokasi diberikan :

$$\text{TotalCost} = \sum_{i=1}^m \sum_{j=1}^n C_{ij} * A_{ij}$$

2.2 Initial Base Feasible Solution (IBFS)

Secara garis besar pada *Transportation Problem*, untuk mendapatkan hasil paling optimal terdapat 2 tahapan. Tahapan yang pertama adalah inisiasi yaitu mencari IBFS dan tahapan kedua adalah optimalisasi yaitu melakukan *Stepping Stone*. IBFS adalah menerapkan sebuah algoritma tertentu pada permasalahan *Transportation Problem* sehingga mendapatkan solusi awal [4]. Setelah menerapkan algoritma IBFS, solusi awal yang dihasilkan tersebut kemudian diproses pada tahap *Stepping Stone*. Umumnya, tahapan *Stepping Stone* memakan waktu yang lama, bahkan waktu yang dibutuhkan untuk tahapan ini tidak senilai walaupun hasil yang didapatkan sudah optimal. Hal ini menyebabkan para peneliti banyak melakukan riset untuk menciptakan algoritma IBFS untuk sedekat mungkin mendapatkan hasil optimal. Algoritma IBFS yang baik adalah algoritma yang dapat menghasilkan solusi dengan biaya yang

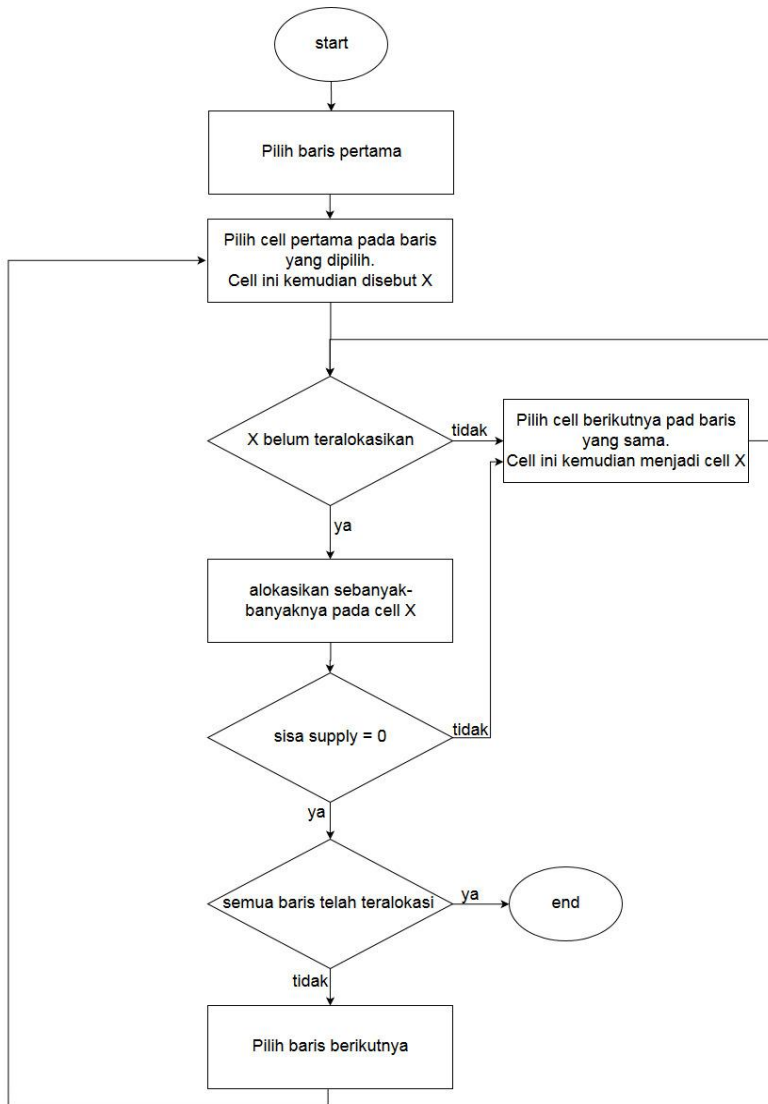
paling kecil. Semakin mendekati angka minimal, semakin baik pula algoritma tersebut [3].

2.3 Algoritma Northwest Corner Method

Salah satu algoritma IBFS yang umum digunakan adalah algoritma *Northwest Corner Method* (NWCM). Pada algoritma ini, alokasi diberikan berurutan pada setiap baris tanpa melakukan perbandingan terhadap *cost* setiap *cell*. Secara garis besar, langkah-langkah pada algoritma NWCM adalah sebagai berikut,

1. Pilih *cell* pada baris pertama kolom pertama,
2. Perhatikan *cell* yang telah dipilih kemudian lakukan langkah dengan memperhatikan percabangan sebagai berikut
 - i. Apabila *cell* masih belum teralokasi, beri alokasi sebanyak-banyaknya di *cell* tersebut
 - ii. Apabila *cell* telah teralokasi, pilih *cell* pada kolom berikutnya pada baris tersebut, kemudian ulangi langkah 2
3. Perhatikan sisa supply pada baris tersebut, kemudian lakukan langkah dengan memperhatikan percabangan sebagai berikut
 - i. Apabila supply masih tersisa, maka pilih *cell* pada kolom berikutnya pada baris tersebut, kemudian kembali ke langkah 2
 - ii. Apabila supply pada baris tersebut telah habis, sedangkan supply pada baris berikutnya masih tersisa, maka pindah ke baris berikutnya. Pilih *cell* pada kolom pertama baris tersebut dan kemudian kembali ke langkah 2
 - iii. Apabila supply pada semua baris telah teralokasikan maka lanjut ke langkah 4.
4. Hitung jumlah biaya pengiriman.

Flowchart algoritma *Northwest Corner Method* dapat dilihat pada gambar 2 berikut.



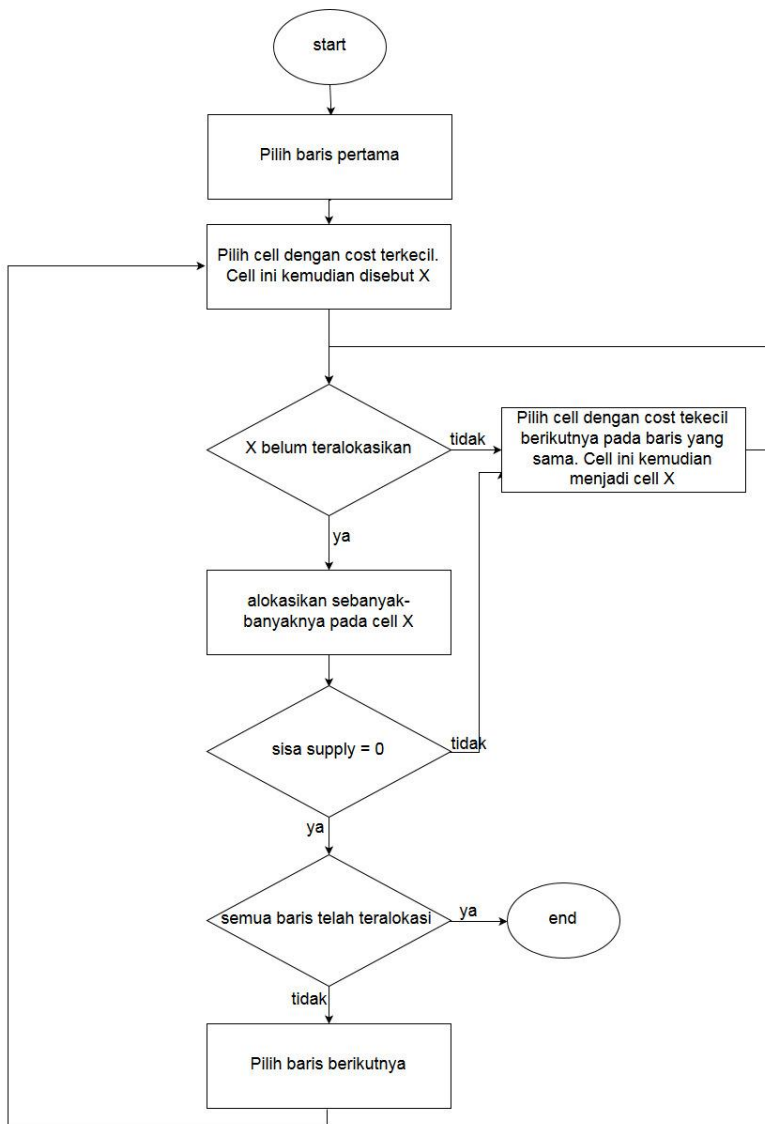
Gambar 2. Flowchart algoritma NWCM

2.4 Algoritma Least Cost Method

Pada beberapa riset, peneliti sepakat bahwa algoritma NWCM kurang efektif untuk mendapatkan hasil yang optimal. Algoritma *Least Cost Method* (LCM) adalah pengembangan dari algoritma NWCM. Pada algoritma LCM, pengalokasian pada setiap baris didasarkan pada perbandingan *cost*. Secara garis besar, langkah-langkah pada algoritma LCM adalah sebagai berikut,

1. Pilih *cell* dengan *cost* terkecil baris pertama
2. Perhatikan *cell* yang telah dipilih kemudian lakukan langkah dengan memperhatikan percabangan sebagai berikut
 - i. Apabila *cell* masih belum teralokasi, beri alokasi sebanyak-banyaknya di *cell* tersebut
 - ii. Apabila *cell* telah teralokasi, pilih *cell* dengan *cost* terkecil berikutnya pada baris tersebut, kemudian ulangi langkah 2
3. Perhatikan sisa supply pada baris tersebut, kemudian lakukan langkah dengan memperhatikan percabangan sebagai berikut
 - i. Apabila supply masih tersisa, maka pilih *cell* dengan *cost* terkecil berikutnya pada baris tersebut, kemudian kembali ke langkah 2
 - ii. Apabila supply pada baris tersebut telah habis, sedangkan supply pada baris berikutnya masih tersisa, maka pindah ke baris berikutnya. Pilih *cell* dengan *cost* terkecil pada baris tersebut dan kemudian kembali ke langkah 2
 - iii. Apabila supply pada semua baris telah teralokasikan maka lanjut ke langkah 4.
4. Hitung jumlah biaya pengiriman.

Flowchart algoritma *Least Cost Method* dapat dilihat pada gambar 3 berikut.



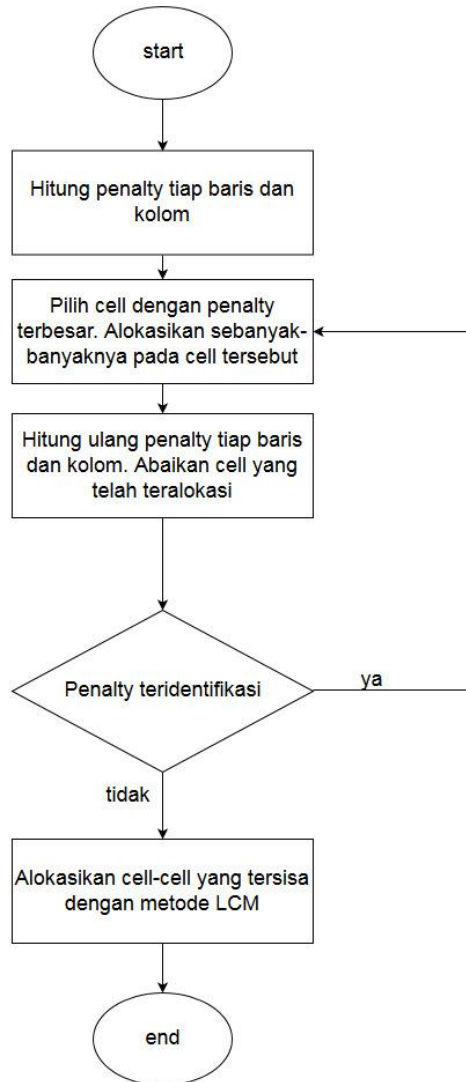
Gambar 3. Flowchart algoritma LCM

2.5 Algoritma Vogel Approximation Method

Pengembangan berikutnya yang diciptakan dari riset para peneliti adalah algoritma *Vogel Approximation Method* (VAM). Pada algoritma VAM, alokasi diberikan dengan membandingkan *penalty* setiap baris dan kolom. Secara garis besar, langkah-langkah pada algoritma VAM adalah sebagai berikut,

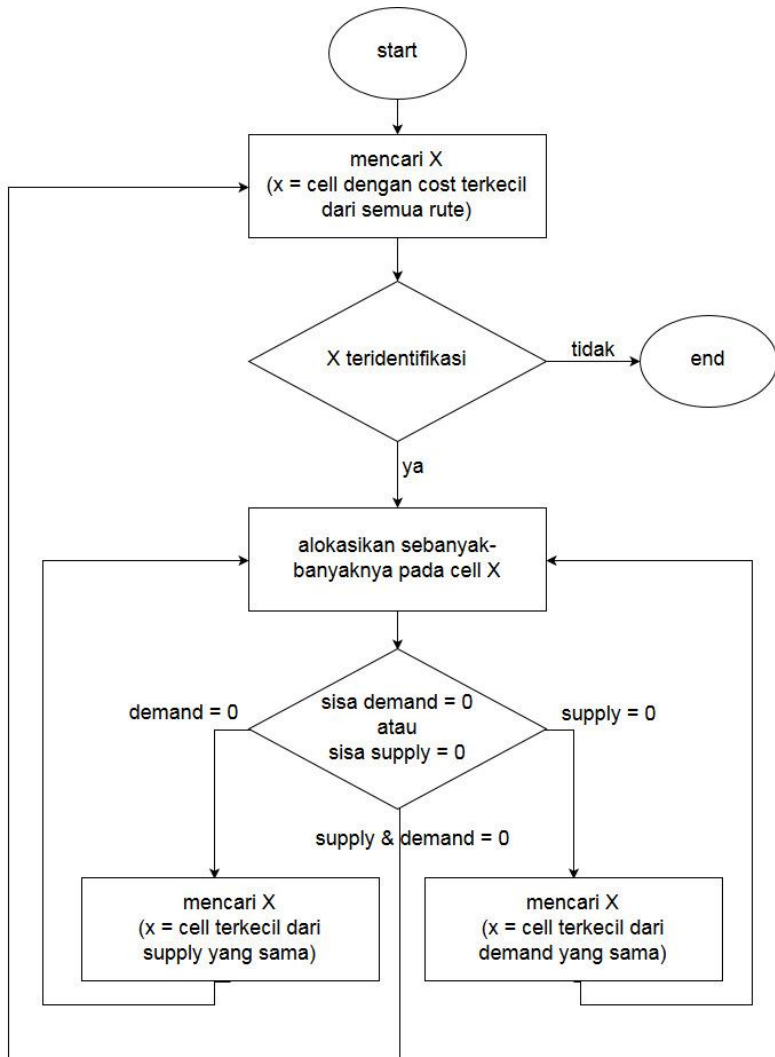
1. Pada setiap baris, cari 2 *cell* dengan *cost* terkecil. Hasil selisih 2 *cell* tersebut dinamakan *penalty* baris.
2. Pada setiap kolom, cari 2 *cell* dengan *cost* terkecil. Hasil selisih 2 *cell* tersebut dinamakan *penalty* kolom.
3. Pilih baris/kolom dengan *penalty* terbesar. Alokasikan sebanyak-banyaknya kepada *cell* tersebut.
4. hitung kembali *penalty* tiap baris dan kolom dengan mengabaikan *cell* yang telah diberikan alokasi. Kemudian perhatikan *cell-cell* yang belum teralokasikan dan lakukan langkah dengan memperhatikan percabangan sebagai berikut
 - i. Apabila terdapat baris/kolom yang hanya menyisakan kurang dari 2 *cell*, maka baris/kolom tersebut tidak perlu dihitung *penalty*-nya. Setelah menghitung *penalty* baris dan kolom yang tersisa, kembali ke langkah 3
 - ii. Apabila semua baris/kolom menyisakan kurang dari 2 *cell*, maka *penalty* tidak perlu dihitung dan lanjutkan ke langkah 5
5. Selesaikan pengalokasian pada *cell-cell* yang tersisa dengan menggunakan metode LCM
6. Apabila semua *cell* telah teralokasikan, hitung jumlah biaya pengiriman.

Flowchart algoritma *Vogel Approximation Method* dapat dilihat pada gambar 4 berikut.



Gambar 4. *Flowchart* algoritma VAM

2.6 Algoritma Incessant Allocation Method



Gambar 5. Flowchart algoritma IAM

Incessant Allocation Method (IAM) adalah salah satu algoritma IBFS yang ditemukan oleh Mollah Mesbahuddin pada tahun 2016 lalu [2]. *Flowchart* IAM dapat dilihat pada gambar 5. Secara garis besar, langkah-langkah dalam algoritma IAM adalah sebagai berikut :

1. Cari *cell* dengan biaya terkecil dari semua *cell*. Untuk selanjutnya *cell* ini dinamakan dengan *cell X*.
2. Apabila semua *cell* telah terisi, lanjut ke langkah 5
3. Lakukan pengalokasian sebanyak mungkin pada *cell X*
4. Perhatikan jumlah *supply* dan *demand* yang tersisa setelah mengalokasikan *cell X*, kemudian lakukan langkah dengan memperhatikan percabangan sebagai berikut
 - i. Apabila jumlah *demand* dan *supply* keduanya habis, maka kembali ke langkah 1
 - ii. Apabila jumlah *supply* masih tersisa, maka cari *cell* dengan biaya terkecil berikutnya yang masih dalam satu *supply*. Kemudian *cell* ini yang sekarang menjadi *cell X*. Kembali ke langkah 2
 - iii. Apabila jumlah *demand* masih tersisa, maka cari *cell* dengan biaya terkecil berikutnya yang masih berada dalam satu *demand*. Kemudian *cell* ini yang sekarang menjadi *cell X*. Kembali ke langkah 2
5. Hitung jumlah biaya pengiriman.

Berikut ini adalah contoh permasalahan *Transportation Problem*. Diketahui sebuah soal *Transportation Problem* dengan notasi sebagai berikut :

$[C_{ij}] \ 3 \times 4 = [1, 6, 5, 9]; [3, 2, 4, 15]; [8, 9, 3, 10]$ $[S_i] \ 3 \times 1 = [20, 25, 15]$ $[D_j] \ 1 \times 4 = [15, 15, 10, 20]$
--

Problem Set 1. Test-case 1 Transportation Problem

Matrix *Transportation Problem* ini dapat dilihat pada tabel 2.

Tabel 2. Matrix Transportation Problem pada Problem Set 1

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1	6	5	9	20
F ₂	3	2	4	15	25
F ₃	8	9	3	10	15
Demand	15	15	10	20	

Langkah pertama pada algoritma IAM adalah mencari *cell* dengan biaya terkecil kemudian mengalokasikan sebanyak-banyaknya. Pada matrix tersebut, biaya terkecil adalah 1 yaitu *cell* dengan koordinat (1,1). Progress pengerjaan soal hingga tahap ini dapat dilihat pada tabel 3

Tahapan ini adalah perulangan pertama. Karena *demand* W₁ telah habis, maka semua *cell* yang menuju kolom W₁ tidak dapat diberikan alokasi. Sedangkan karena *supply* F₁ masih tersisa, maka algoritma IAM akan berlanjut pada *cell-cell* pada baris F₁. *Cell* dengan biaya terkecil pada baris F₁ adalah *cell* (1, 3) dengan biaya 5. Alokasikan sebanyak-banyaknya pada *cell* tersebut. Progress pengerjaan soal hingga tahap ini dapat dilihat pada tabel 4.

Tahapan ini adalah perulangan kedua. Karena *supply* F₁ telah habis, maka semua *cell* yang menuju baris F₁ tidak dapat diberikan alokasi. Sedangkan karena *demand* W₃ masih tersisa,

maka algoritma IAM akan berlanjut pada *cell-cell* pada kolom W_3 . *Cell* dengan biaya terkecil pada kolom W_3 adalah *cell* (3, 3) dengan biaya 3. Alokasikan sebanyak-banyaknya pada *cell* tersebut. Progress pengerjaan soal hingga tahap ini dapat dilihat pada tabel 5.

Tahapan ini adalah perulangan ketiga. Karena *demand* W_3 telah habis, maka semua *cell* yang menuju kolom W_3 tidak dapat diberikan alokasi. Sedangkan karena *supply* F_3 masih tersisa, maka algoritma IAM akan berlanjut pada *cell-cell* pada baris F_3 . *Cell* dengan biaya terkecil pada baris F_3 adalah *cell* (3, 2) dengan biaya 9. Alokasikan sebanyak-banyaknya pada *cell* tersebut. Progress pengerjaan soal hingga tahap ini dapat dilihat pada tabel 6.

Tahapan ini adalah perulangan keempat. Karena *supply* F_3 telah habis, maka semua *cell* yang menuju baris F_3 tidak dapat diberikan alokasi. Sedangkan karena *demand* W_2 masih tersisa, maka algoritma IAM akan berlanjut pada *cell-cell* pada kolom W_2 . *Cell* dengan biaya terkecil pada kolom W_2 adalah *cell* (2, 2) dengan biaya 2. Alokasikan sebanyak-banyaknya pada *cell* tersebut. Progress pengerjaan soal hingga tahap ini dapat dilihat pada tabel 7.

Tahapan ini adalah perulangan kelima. Karena *supply* F_2 masih tersisa, maka algoritma IAM akan berlanjut pada *cell-cell* pada baris F_2 . *Cell* dengan biaya terkecil pada baris F_2 adalah *cell* (2, 4) dengan biaya 15. Alokasikan sebanyak-banyaknya pada *cell* tersebut. Progress pengerjaan soal hingga tahap ini dapat dilihat pada tabel 8.

Pada tahap ini algoritma IAM telah selesai. Total biaya yang didapatkan dari algoritma IAM adalah sebagai berikut :

$$\begin{aligned}\text{Total Biaya} &= (1 \times 15) + (5 \times 5) + (2 \times 5) + (15 \times 20) + \\ &\quad (9 \times 10) + (3 \times 5) \\ &= 455\end{aligned}$$

Tabel 3. Tabel solusi algoritma IAM tahap inisiasi

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1	6	5	9	5
	15				
F ₂	3	2	4	15	25
	0				
F ₃	8	9	3	10	15
	0				
Demand	0	15	10	20	

Tabel 4. Tabel solusi algoritma IAM tahap perulangan I

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1	6	5	9	0
	15	0	5	0	
F ₂	3	2	4	15	25
	0				
F ₃	8	9	3	10	15
	0				
Demand	0	15	5	20	

Tabel 5. Tabel solusi algoritma IAM tahap perulangan II

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1 15	6 0	5 5	9 0	0
F ₂	3 0	2	4 0	15	25
F ₃	8 0	9	3 5	10	10
Demand	0	15	0	20	

Tabel 6. Tabel solusi algoritma IAM tahap perulangan III

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1 15	6 0	5 5	9 0	0
F ₂	3 0	2	4 0	15	25
F ₃	8 0	9 10	3 5	10	0
Demand	0	5	0	20	

Tabel 7. Tabel solusi algoritma IAM tahap perulangan IV

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1	6	5	9	0
	15	0	5	0	
F ₂	3	2	4	15	20
	0	5	0		
F ₃	8	9	3	10	0
	0	10	5		
Demand	0	0	0	20	

Tabel 8. Tabel solusi algoritma IAM tahap perulangan V

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1	6	5	9	0
	15	0	5	0	
F ₂	3	2	4	15	0
	0	5	0	20	
F ₃	8	9	3	10	0
	0	10	5	0	
Demand	0	0	0	0	

BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini akan menjelaskan tentang analisis dan perancangan sistem untuk mencapai tujuan dari tugas akhir. Perancangan ini meliputi hal-hal yang akan dimodifikasi pada algoritma IAM.

3.1 Analisis Algoritma IAM

Berikut ini adalah jawaban optimal soal *Problem Set-1* yang telah dibahas pada bab sebelumnya. Jawaban optimal ini didapatkan dengan menggunakan *tools* ToraSystem.

Tabel 9. Tabel solusi optimal Problem Set-1

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1 5	6 0	5 0	9 15	0
F ₂	3 10	2 15	4 0	15 0	0
F ₃	8 0	9 0	3 10	10 5	0
Demand	0	0	0	0	

Total biaya yang didapatkan dari jawaban optimal adalah sebagai berikut :

$$\begin{aligned}
 \text{Total Biaya} &= (1 \times 5) + (9 \times 15) + (3 \times 10) + (2 \times 15) + \\
 &\quad (3 \times 10) + (10 \times 5) \\
 &= 280
 \end{aligned}$$

Dapat dilihat perbedaan yang signifikan terhadap hasil dari jawaban optimal dengan hasil yang didapat dari algoritma IAM. Letak permasalahan utama pada algoritma IAM adalah ketidak-fleksibelannya ketika melakukan alokasi. Dengan menggunakan algoritma IAM, pengalokasian harus dilakukan pada *supply/demand* yang tersisa dari pengalokasian sebelumnya sehingga terkadang banyak *cell* lain yang terabaikan padahal memiliki biaya yang relatif kecil. Misalkan pada *Problem Set-1* pada perulangan II dari algoritma IAM berikut.

Tabel 10. Tabel solusi algoritma IAM tahap perulangan II

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1	6	5	9	0
	15	0	5	0	
F ₂	3	2	4	15	25
	0		0		
F ₃	8	9	3	10	10
	0		5		
Demand	0	15	0	20	

Menurut aturan algoritma IAM, setelah melakukan alokasi pada *cell* (1,3), maka pengalokasian berikutnya harus dilakukan terhadap *cell-cell* pada kolom W₃. Padahal, *cell* (2,2) memiliki biaya yang lebih kecil dan seharusnya diprioritaskan untuk diberi alokasi terlebih dahulu. Hal ini mempunyai pengaruh besar terhadap hasil akhir yang didapatkan. Sehingga pada sub-bab ini dapat disimpulkan bahwa algoritma IAM masih belum efektif.

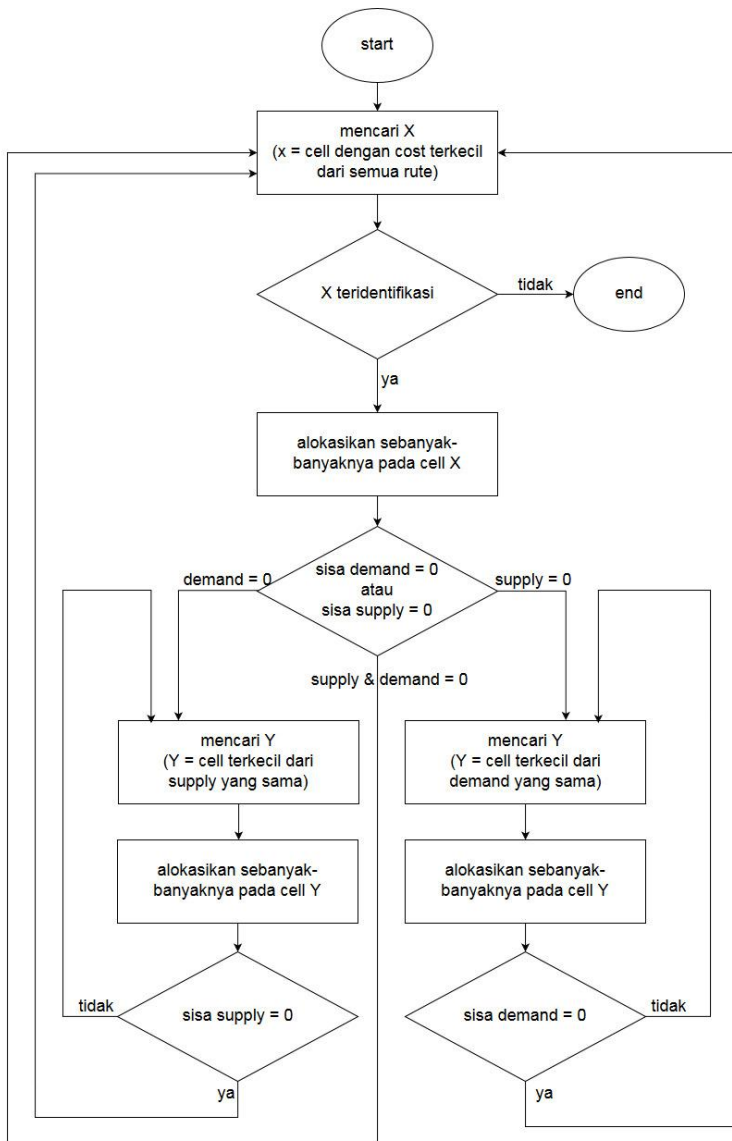
3.2 Perubahan *Looping* pada Inisiasi

Pada sub-bab sebelumnya telah diketahui bahwa kelemahan algoritma IAM adalah kurang fleksibel dalam melakukan alokasi. Oleh karena itu, agar algoritma IAM memiliki akurasi lebih tinggi lagi, algoritma IAM harus diubah dalam melakukan setiap perulangannya.

Pada algoritma IAM, pengalokasian yang dilakukan terikat pada *supply/demand* yang tersisa dari pengalokasian sebelumnya. Hal ini membuat *cell* lain yang memiliki biaya kecil dan berpotensi meminimalisir biaya total tidak dapat terpilih untuk dialokasikan terlebih dahulu.

Algoritma ini kemudian dinamakan *Incessant Allocation Method – Initiation Looping* (IAM-IL) Pada algoritma ini, modifikasi dilakukan pada tahap perulangannya. Secara garis besar, langkah-langkah pada algoritma IAM-IL adalah sebagai berikut :

1. Cari *cell* dengan biaya terkecil dari semua *cell*. Untuk selanjutnya *cell* ini dinamakan dengan *cell* X. Apabila tidak ditemukan *cell* kosong yang tersisa maka lanjut ke langkah 4
2. Alokasikan sebanyak mungkin pada *cell* X
3. Perhatikan jumlah *supply* dan *demand* yang tersisa setelah mengalokasikan *cell* X, kemudian lakukan langkah dengan memperhatikan percabangan sebagai berikut
 - i. Apabila jumlah *demand* dan *supply* keduanya habis, maka lanjut ke langkah 1
 - ii. Apabila jumlah *supply* masih tersisa, maka alokasikan pada *cell* dengan biaya terkecil berikutnya yang masih dalam satu *supply*.
 - iii. Apabila jumlah *demand* masih tersisa, maka alokasikan pada *cell* dengan biaya terkecil berikutnya yang masih berada dalam satu *demand*.
4. Ulangi ke langkah 1
5. Hitung jumlah biaya pengiriman.



Gambar 6. Flowchart algoritma IAM-IL

Flowchart algoritma IAM-IL dapat dilihat pada gambar 6. Berikut ini adalah contoh penerapan algoritma IAM-IL terhadap *Problem Set-1*

Langkah pertama pada algoritma IAM-IL adalah mencari *cell* dengan biaya terkecil kemudian mengalokasikan sebanyak-banyaknya. Pada matrix tersebut, biaya terkecil adalah 1 yaitu *cell* dengan koordinat (1,1). Progress pengerjaan soal ini dapat dilihat pada tabel 11

Tahapan ini adalah perulangan pertama. Karena *demand* W_1 telah habis, maka semua *cell* yang menuju kolom W_1 tidak dapat diberikan alokasi. Sedangkan karena *supply* F_1 masih tersisa, maka algoritma IAM-IL akan berlanjut pada *cell-cell* pada baris F_1 . *Cell* dengan biaya terkecil pada baris F_1 adalah *cell* (1, 3) dengan biaya 5. Alokasikan sebanyak-banyaknya pada *cell* tersebut. Progress pengerjaan soal ini dapat dilihat pada tabel 12.

Tahapan ini adalah perulangan kedua. Karena *supply* 1 telah habis, maka semua *cell* yang menuju *supply* F_1 tidak dapat diberikan alokasi. Langkah berikutnya pada algoritma IAM-IL adalah mencari *cell* dengan biaya terkecil lainnya dari semua *cell/rute*, kemudian mengalokasikan sebanyak-banyaknya. Pada matrix tersebut, biaya terkecil adalah 2 yaitu *cell* dengan koordinat (2,2). Progress pengerjaan soal ini dapat dilihat pada tabel 13.

Tahapan ini adalah perulangan ketiga. Karena *demand* W_2 telah habis, maka semua *cell* yang menuju kolom W_2 tidak dapat diberikan alokasi. Sedangkan karena *supply* F_2 masih tersisa, maka algoritma IAM-IL akan berlanjut pada *cell-cell* pada baris F_2 . *Cell* dengan biaya terkecil pada baris F_2 adalah *cell* (2, 3) dengan biaya 4. Alokasikan sebanyak-banyaknya pada *cell* tersebut. Progress pengerjaan soal ini dapat dilihat pada tabel 14.

Tahapan ini adalah perulangan keempat. Karena *supply* F_2 belum habis, maka algoritma IAM-IL masih berlanjut pada *cell-cell* pada baris F_2 . *Cell* dengan biaya terkecil pada baris F_2 adalah *cell* (2, 4) dengan biaya 15. Alokasikan sebanyak-banyaknya pada

cell tersebut. Progress pengerjaan soal ini dapat dilihat pada tabel 15.

Tahapan ini adalah perulangan kelima. Karena *supply* F_2 telah habis, maka semua *cell* yang menuju baris F_2 tidak dapat diberikan alokasi. Langkah berikutnya pada algoritma IAM-IL adalah mencari *cell* dengan biaya terkecil lainnya kemudian mengalokasikan sebanyak-banyaknya. Pada matrix tersebut, biaya terkecil adalah 10 yaitu *cell* dengan koordinat (3,4). Progress pengerjaan soal ini dapat dilihat pada tabel 16.

Tabel 11. Tabel solusi algoritma IAM-IL tahap inisiasi

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1	6	5	9	5
	15				
F ₂	3	2	4	15	25
	0				
F ₃	8	9	3	10	15
	0				
Demand	0	15	10	20	

Tabel 12. Tabel solusi algoritma IAM-IL tahap perulangan I

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1 15	6 0	5 5	9 0	0
F ₂	3 0	2	4	15	25
F ₃	8 0	9	3	10	15
Demand	0	15	5	20	

Tabel 13. Tabel solusi algoritma IAM-IL tahap perulangan II

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1 15	6 0	5 5	9 0	0
F ₂	3 0	2 15	4	15	10
F ₃	8 0	9 0	3	10	10
Demand	0	0	5	20	

Tabel 14. Tabel solusi algoritma IAM-IL tahap perulangan III

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1	6	5	9	0
	15	0	5	0	
F ₂	3	2	4	15	5
	0	15	5		
F ₃	8	9	3	10	0
	0	0	0		
Demand	0	0	0	20	

Tabel 15. Tabel solusi algoritma IAM-IL tahap perulangan IV

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1	6	5	9	0
	15	0	5	0	
F ₂	3	2	4	15	0
	0	15	5	5	
F ₃	8	9	3	10	0
	0	0	0		
Demand	0	0	0	15	

Tabel 16. Tabel solusi algoritma IAM-IL tahap perulangan V

Factories	Warehouses								Supply
	W ₁		W ₂		W ₃		W ₄		
F ₁	1		6000		5		9		0
	15		0		5		0		
F ₂	3		2		4		15		0
	0		15		5		5		
F ₃	8		9		3		10		0
	0		0		0		15		
Demand	0		0		0		0		

Pada tahap ini algoritma IAM-IL telah selesai. Total biaya yang didapatkan dari algoritma IAM-IL adalah sebagai berikut :

$$\begin{aligned}
 \text{Total Biaya} &= (1 \times 15) + (5 \times 5) + (2 \times 15) + (4 \times 5) + \\
 &\quad (15 \times 5) + (10 \times 15) \\
 &= 315
 \end{aligned}$$

Hasil dari algoritma IAM-IL lebih kecil dan lebih mendekati nilai jawaban optimal daripada hasil dari algoritma IAM asli.

3.3 Perbandingan Prioritas pada Cell

Sub-bab ini akan membahas mengenai pengembangan lebih lanjut algoritma IAM. Pada sub-bab sebelumnya, algoritma IAM-IL masih belum 100% akurat meskipun besarnya error sudah terminimalisir. Pengembangan selanjutnya adalah pada prioritas pemilihan *cell* yang akan dialokasikan.

Tabel 17. Tabel solusi optimal Problem Set-1

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1	6	5	9	0
	5	0	0	15	
F ₂	3	2	4	15	0
	10	15	0	0	
F ₃	8	9	3	10	0
	0	0	10	5	
Demand	0	0	0	0	

Pada tabel solusi optimal *Problem Set-1* berikut ini dapat dilihat bahwa *cell* (1,1) hanya diberikan alokasi sebesar 5. Hal ini membuktikan bahwa prioritas utama alokasi pada *problem-set* ini seharusnya tidak terletak pada *cell* (1,1), padahal pada algoritma IAM maupun IAM-IL, alokasi pertama pasti akan diberikan pada *cell* (1,1) yang memiliki *cost* terkecil yaitu 1. Dengan demikian, hal ini membuktikan bahwa terdapat faktor lain yang lebih mempengaruhi penting-tidaknya sebuah *cell* untuk dialokasikan terlebih dahulu selain besarnya nilai *cost*.

Faktor lain yang dapat mempengaruhi prioritas pengalokasian adalah perbandingan terhadap *cell* lain dalam baris/ kolom yang sama. Metode ini disebut perbandingan prioritas. Tujuan metode ini adalah mencari *priority value* setiap *cell* agar dapat mengetahui *cell* mana yang seharusnya dialokasikan terlebih dahulu. Berikut ini adalah rumus untuk menghitung *priority value* setiap *cell*.

$$\text{RowPriorityValue}(x,y) = \sum_{n=1}^i \text{cost}(i,y) - n * \text{cost}(x,y)$$

$$\text{ColumnPriorityValue}(x,y) = \sum_{n=1}^j \text{cost}(x,j) - n * \text{cost}(x,y)$$

x,y = koordinat *cell* yang akan dihitung *priority value*-nya
 n = banyaknya jumlah *supply/demand*

Semakin besar nilai *priority value* maka semakin penting pula *cell* tersebut dialokasikan terlebih dahulu. Metode perbandingan prioritas ini dapat diterapkan pada algoritma IAM-IL untuk mendapat hasil yang lebih baik. Algoritma yang baru ini kemudian dinamakan algoritma *Incessant Allocation Method – Priority Value* (IAM-PV). Secara garis besar, langkah-langkah pada algoritma IAM-PV adalah sebagai berikut :

1. Hitung *row priority value* dan *column priority value* semua *cell* yang masih kosong. Apabila semua *cell* telah terisi maka lanjut ke langkah 6
2. Cari *cell* dengan *priority value* terbesar dari semua *cell*. Untuk selanjutnya *cell* ini dinamakan dengan *cell X*.
3. Alokasikan sebanyak mungkin pada *cell X*
4. Perhatikan jumlah *supply* dan *demand* yang tersisa setelah mengalokasikan *cell X*, kemudian lakukan langkah dengan memperhatikan percabangan sebagai berikut
 - i. Apabila jumlah *demand* dan *supply* keduanya habis, maka kembali ke langkah 1
 - ii. Apabila jumlah *supply* masih tersisa, maka cari *cell* dengan *priority value* terbesar berikutnya dalam satu baris

- iii. Apabila jumlah *demand* masih tersisa, maka cari *cell* dengan *priority value* terbesar berikutnya dalam satu kolom
5. Kembali ke langkah 1
6. Hitung jumlah biaya pengiriman.

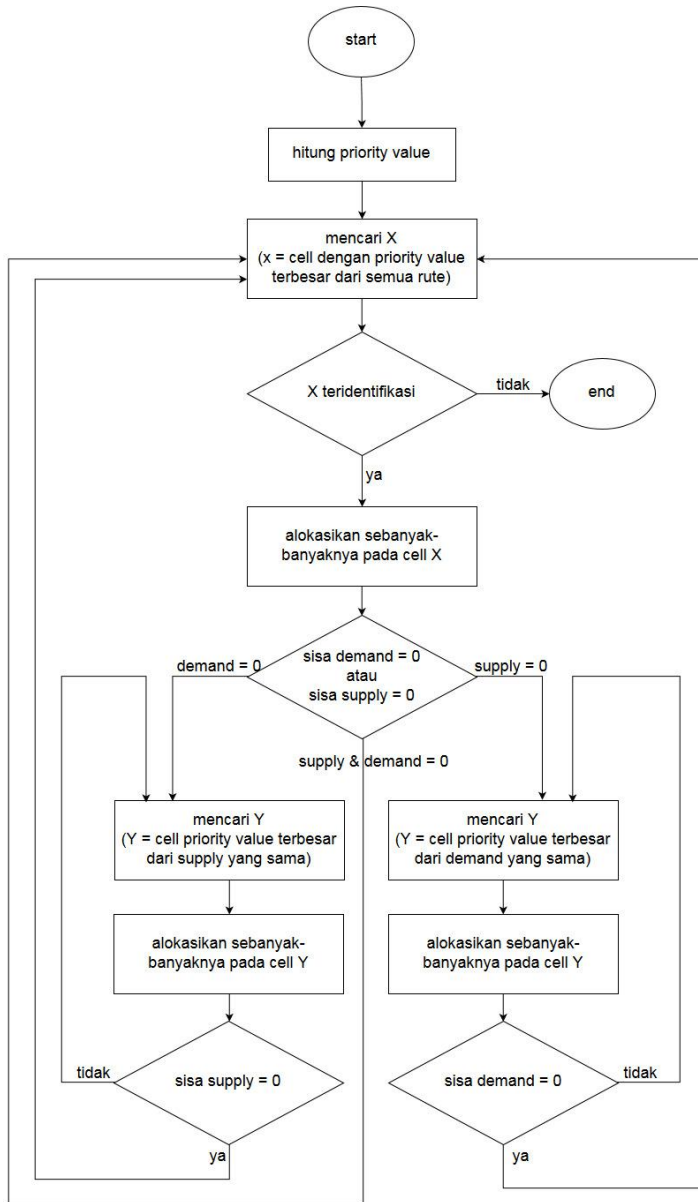
Flowchart algoritma IAM-PV dapat dilihat pada gambar 7. Berikut ini adalah contoh penerapan algoritma IAM-PV terhadap *Problem Set-1*

Langkah pertama pada algoritma IAM-PV adalah mencari *cell* dengan *priority value* terbesar. Tabel *priority value* pada tahap ini dapat dilihat pada tabel 18. Pada matrix tersebut, *priority value* terbesar adalah *cell* dengan koordinat (3,3). Alokasikan kepada *cell* (3,3). Selanjutnya karena *supply* F_3 masih tersisa, maka dialokasikan pada *cell* dengan *priority value* terbesar pada baris F_3 yaitu *cell* dengan koordinat (3,4). Progress pengerjaan soal ini dapat dilihat pada tabel 19

Tahapan ini adalah perulangan pertama. Cari *cell* dengan *priority value* terbesar berikutnya. Tabel *priority value* pada tahap ini dapat dilihat pada tabel 20. Pada matrix tersebut, *priority value* terbesar adalah *cell* dengan koordinat (2,2). Alokasikan kepada *cell* (2,2). Selanjutnya karena *supply* F_2 masih tersisa, maka dialokasikan pada *cell* dengan *priority value* terbesar pada baris F_2 yaitu *cell* dengan koordinat (2,1). Progress pengerjaan soal ini dapat dilihat pada tabel 21

Tahapan ini adalah perulangan kedua. Cari *cell* dengan *priority value* terbesar berikutnya. Tabel *priority value* pada tahap ini dapat dilihat pada tabel 22. Pada matrix tersebut, *priority value* terbesar adalah *cell* dengan koordinat (1,1). Karena semua *supply* telah habis maka langkah selanjutnya adalah menghitung total biaya. Progress pengerjaan soal ini dapat dilihat pada tabel 23

$$\begin{aligned}
 \text{Total Biaya} &= (1 \times 5) + (9 \times 15) + (3 \times 10) + (2 \times 15) + \\
 &\quad (3 \times 10) + (10 \times 5) \\
 &= 280
 \end{aligned}$$



Gambar 7. Flowchart algoritma IAM-PV

Tabel 18. Tabel *priority value* tahap inisiasi

Factories		Warehouses			
		W ₁	W ₂	W ₃	W ₄
F ₁	RPV	17	-3	1	-15
	CPV	9	-1	-3	7
F ₂	RPV	12	16	8	-36
	CPV	3	11	0	-11
F ₃	RPV	-2	-6	18	-10
	RPV	-12	-10	3	4

Tabel 19. Tabel solusi algoritma IAM-PV tahap inisiasi

Factories	Warehouses								Supply
	W ₁		W ₂		W ₃		W ₄		
F ₁	1		6		5		9		5
					0				
F ₂	3		2		4		15		25
					0				
F ₃	8		9		3		10		0
	0		0		10		5		
Demand	15		15		0		15		

Tabel 20. Tabel *priority value* tahap perulangan I

Factories		Warehouses			
		W_1	W_2	W_3	W_4
F_1	RPV	13	-2		-11
	CPV	2	-4		6
F_2	RPV	11	14		-25
	CPV	-2	4		-6
F_3	RPV				
	RPV				

Tabel 21. Tabel solusi algoritma IAM-PV tahap perulangan I

Factories	Warehouses				Supply
	W_1	W_2	W_3	W_4	
F_1	1	6	5	9	0
		0	0		
F_2	3	2	4	15	0
	10	15	0	0	
F_3	8	9	3	10	0
	0	0	10	5	
Demand	5	0	0	15	

Tabel 22. Tabel *priority value* tahap perulangan II

Factories		Warehouses			
		W ₁	W ₂	W ₃	W ₄
F ₁	RPV	8			-8
	CPV	0			0
F ₂	RPV				
	CPV				
F ₃	RPV				
	RPV				

Tabel 23. Tabel solusi algortima IAM-PV tahap perulangan II

Factories	Warehouses				Supply
	W ₁	W ₂	W ₃	W ₄	
F ₁	1	6	5	9	0
	5	0	0	15	
F ₂	3	2	4	15	0
	10	15	0	0	
F ₃	8	9	3	10	0
	0	0	10	5	
Demand	0	0	5	0	

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari perancangan sistem sesuai dengan perancangan yang telah dibuat. Bahasa pemrograman yang digunakan untuk implementasi sistem adalah Bahasa pemrograman C.

4.1 Lingkungan Implementasi

Lingkungan implementasi sistem yang digunakan untuk mengembangkan tugas akhir ini memiliki spesifikasi perangkat keras dan perangkat lunak yang ditunjukkan oleh Tabel 24.

Tabel 24. Spesifikasi Perangkat

Perangkat	Spesifikasi
Perangkat Keras	<ul style="list-style-type: none">• Prosesor: Intel® Core™ i7-7500U CPU @ 2.70GHz (4 CPUs), ~2.9GHz• Memori: 8192MB
Perangkat Lunak	<ul style="list-style-type: none">• Sistem Operasi Microsoft Windows 7 64-bit• Perangkat Pengembang Dev C++ 5.4.1• Perangkat Pembantu Notepad, Microsoft Excel 2007, Microsoft Word 2016, Microsoft Power Point 2016, Snipping Tools

4.2 Implementasi Algoritma IAM-PV

Implementasi proses dilakukan berdasarkan perancangan proses yang dijelaskan pada bab analisis dan perancangan. Algoritma yang akan diimplementasikan adalah algoritma IAM-PV. Algoritma IAM-PV akan diimplementasikan dalam bentuk program berbahasa C dengan menggunakan *tools* Dev-C++ 5.4.1

4.2.1. Implementasi Tahap Menghitung Priority Value

Bagian ini membahas implementasi algoritma IAM-PV pada tahapan menghitung *Row Priority Value* dan *Column Priority Value*. Implementasi pada Kode Sumber 1. menunjukkan pengimplementasian dari algoritma IAM-PV pada tahap menghitung *Priority Value*

```

1.  for(j=0;j<jumlah_demand;j++)
2.  {
3.      int sum=0;
4.      for(i=0;i<jumlah_supply;i++)
5.          {if(allocation[i][j]==-1)
6.              sum+=cost[i][j]}
7.      for(i=0;i<jumlah_supply;i++)
8.          {if(allocation[i][j]==-1)
9.              {v_cost[i][j]=sum-
10.                 cost[i][j]*(jumlah_supply - supply_satisfy)}}
11. }
12.
13. for(i=0;i<jumlah_supply;i++)
14. {
15.     int sum=0;
16.     for(j=0;j<jumlah_demand;j++)
17.         {if(allocation[i][j]==-1)
18.             sum+=cost[i][j];}
19.     for(j=0;j<jumlah_demand;j++)
20.         {if(allocation[i][j]==-1)
21.             {h_cost[i][j]=sum-
22.                 cost[i][j]*(jumlah_demand - demand_satisfy)}}
23. }
```

Kode Sumber 1 Implementasi tahap menghitung Priority Value

4.2.2. Implementasi Tahap Memilih Priority Value

Bagian ini membahas implementasi algoritma IAM-PV pada tahapan memilih *Priority Value* terbesar Implementasi pada Kode Sumber 2. menunjukkan implementasi dari algoritma IAM-PV pada tahap memilih *Priority Value* terbesar.

```

1. int max_penalty=-100000;
2. int max_penalty_i, max_penalty_j=-1;
3. for(i=0;i<jumlah_supply;i++)
4. {
5.     for(j=0;j<jumlah_demand;j++)
6.     {
7.         if(v_cost[i][j]>max_penalty && allocation
           [i][j]==-1)
8.         {
9.             max_penalty=v_cost[i][j];
10.            max_penalty_i=i;
11.            max_penalty_j=j;
12.        }
13.
14.        if(h_cost[i][j]>max_penalty && allocation
           [i][j]==-1)
15.        {
16.            max_penalty=h_cost[i][j];
17.            max_penalty_i=i;
18.            max_penalty_j=j;
19.        }
20.    }
21. }

```

Kode Sumber 2 Implementasi tahap mencari Priority Value terbesar

4.2.3. Implementasi Tahap Alokasi Supply Demand Habis

Bagian ini membahas implementasi algoritma IAM-PV pada tahapan alokasi. Implementasi pada Kode Sumber 3. menunjukkan implementasi dari algoritma IAM-PV pada tahap pengalokasian pada kondisi *supply* dan *demand* habis.

```

1. if(supply[max_penalty_i]==demand[max_penalty_j])
2. {
3.     allocation[max_penalty_i][max_penalty_j]=
       demand[max_penalty_j];
4.     supply[max_penalty_i]=0;
5.     demand[max_penalty_j]=0;

```

```

6.     for(i=0;i<jumlah_supply;i++)
7.         if(allocation[i][max_penalty_j]==-1)
8.             allocation[i][max_penalty_j]=0;
9.     for(j=0;j<jumlah_demand;j++)
10.        if(allocation[max_penalty_i][j]==-1)
11.            allocation[max_penalty_i][j]=0;
12.    supply_satisfy++;
13.    demand_satisfy++;
14. }

```

Kode Sumber 3 Implementasi tahap alokasi supply demand habis

4.2.4. Implementasi Tahap Alokasi Supply Habis

Bagian ini membahas implementasi algoritma IAM-PV pada tahapan alokasi. Implementasi pada Kode Sumber 5. menunjukkan pengimplementasian dari algoritma IAM-PV pada tahap pengalokasian dalam kondisi *supply* habis.

```

1. if(supply[max_penalty_i]<demand[max_penalty_j])
2. {
3.     allocation[max_penalty_i][max_penalty_j]= supply[max_penalty_i];
4.     demand[max_penalty_j]-= supply[max_penalty_i];
5.     supply[max_penalty_i]= 0;
6.     for(j=0;j<jumlah_demand;j++)
7.         if(allocation[max_penalty_i][j]==-1)
8.             allocation[max_penalty_i][j]=0;
9.     supply_satisfy++;
10.
11. while(demand[max_penalty_j]>0)
12. {
13.     int next_penalty=-100000;
14.     int next_penalty_i=-1;
15.     for(i=0;i<jumlah_supply;i++)
16.     {
17.         if(v_cost[i][max_penalty_j]>next_penalty
18.         && allocation[i][max_penalty_j]==-1)
19.             next_penalty= v_cost[i][max_penalty_j];

```

```

19.         next_penalty_i=i;
20.
21.         if(h_cost[i][max_penalty_j]>next_penalty
    && allocation[i][max_penalty_j]==-1)
22.             next_penalty= h_cost[i][max_penalty_j];
23.             next_penalty_i=i;
24.         }
25.     }
26.
27. if(supply[next_penalty_i]>demand[max_penalty_j]
28. {
29.     allocation[next_penalty_i][max_penalty_j]=
    demand[max_penalty_j];
30.     supply[next_penalty_i]-
    = demand[max_penalty_j];
31.     demand[max_penalty_j]= 0;
32.     for(i=0;i<jumlah_supply;i++)
33.         if(allocation[i][max_penalty_j]==-1)
34.             allocation[i][max_penalty_j]=0;
35.             demand_satisfy++;
36.
37.     else if(supply[next_penalty_i]<
    demand[max_penalty_j])
38.         allocation[next_penalty_i][max_penalty_j]
    = supply[next_penalty_i];
39.         demand[max_penalty_j]-
    = supply[next_penalty_i];
40.         supply[next_penalty_i]= 0;
41.         for(j=0;j<jumlah_demand;j++)
42.             if(allocation[next_penalty_i][j]==-1)
43.                 allocation[next_penalty_i][j] =0;
44.                 supply_satisfy++;
45.
46.     else if(supply[next_penalty_i]==
    demand[max_penalty_j])
47.     {
48.         allocation[next_penalty_i][max_penalty_j]
    = supply[next_penalty_i];

```

```

49.         demand[max_penalty_j]=0;
50.         supply[next_penalty_i]= 0;
51.         for(i=0;i<jumlah_supply;i++)
52.             if(allocation[i][max_penalty_j]==-1)
53.                 allocation[i][max_penalty_j]=0;

54.         for(j=0;j<jumlah_demand;j++)
55.             if(allocation[next_penalty_i][j]==-1)
56.                 allocation[next_penalty_i][j]=0;

57.         supply_satisfy++;
58.         demand_satisfy++;
59.     }
60. }
61. }

```

Kode Sumber 4 Implementasi tahap alokasi supply habis

4.2.5. Implementasi Tahap Alokasi Demand Habis

Bagian ini membahas implementasi algoritma IAM-PV pada tahapan alokasi. Implementasi pada Kode Sumber 5. menunjukkan pengimplementasian dari algoritma IAM-PV pada tahap pengalokasian dalam kondisi *demand* habis.

```

1. if(supply[max_penalty_i]>demand[max_penalty_j])
2. {
3.     allocation[max_penalty_i][max_penalty_j]=
4.     demand[max_penalty_j];
5.     supply[max_penalty_i]-=
6.     demand[max_penalty_j];
7.     demand[max_penalty_j]= 0;
8.     for(i=0;i<jumlah_supply;i++)
9.         if(allocation[i][max_penalty_j]==1)
10.             allocation[i][max_penalty_j]=0;
11.     demand_satisfy++;
12.     while(supply[max_penalty_i]>0)
13.     {
14.         int next_penalty=-100000;

```

```

14.         int next_penalty_j=-1;
15.         for(j=0;j<jumlah_demand;j++)
16.         {
17.             if(v_cost[max_penalty_i][j]>
next_penalty&&allocation[max_penalty_i][j]==-1)
18.                 next_penalty=v_cost[max_penalty_i][j];
19.                 next_penalty_j=j;
20.
21.             if(h_cost[max_penalty_i][j]>
next_penalty&&allocation[max_penalty_i][j]==-1)
22.                 next_penalty=h_cost[max_penalty_i][j];
23.                 next_penalty_j=j;
24.         }
25.
26.         if(supply[max_penalty_i]>
demand[next_penalty_j])
27.         {
28.             allocation[max_penalty_i][next_penalty_j]
=demand[next_penalty_j];
29.             supply[max_penalty_i]-
=demand[next_penalty_j];
30.             demand[next_penalty_j]= 0;
31.             for(i=0;i<jumlah_supply;i++)
32.                 if(allocation[i][next_penalty_j]==-1)
33.                     allocation[i][next_penalty_j]=0;
34.             demand_satisfy++;
35.         }
36.         else if(supply[max_penalty_i]<
demand[next_penalty_j])
37.         {
38.             allocation[max_penalty_i][next_penalty_j]
=supply[max_penalty_i];
39.             demand[next_penalty_j]-
=supply[max_penalty_i];
40.             supply[max_penalty_i]= 0;
41.             for(j=0;j<jumlah_demand;j++)
42.                 if(allocation[max_penalty_i][j]==-1)
43.                     allocation[max_penalty_i][j]=0;

```

```

44.         supply_satisfy++;
45.     }
46.     else if(supply[max_penalty_i]==demand
[next_penalty_j])
47.     {
48.         allocation[max_penalty_i][next_penalty_j]
=supply[max_penalty_i];
49.         demand[next_penalty_j]=0;
50.         supply[max_penalty_i]= 0;
51.         for(i=0;i<jumlah_supply;i++)
52.         if(allocation[i][next_penalty_j]==-1)
53.             allocation[i][next_penalty_j]=0;
54.         for(j=0;j<jumlah_demand;j++)
55.             if(allocation[max_penalty_i][j]==-1)
56.                 allocation[max_penalty_i][j]=0;
57.         supply_satisfy++;
58.         demand_satisfy++;
59.     }
60. }
61. }

```

Kode Sumber 5 Implementasi tahap alokasi demand habis

4.2.6. Implementasi Tahap Menghitung Total Cost

Bagian ini membahas implementasi algoritma IAM-PV pada tahapan akhir yaitu menghitung *total cost*. Implementasi pada Kode Sumber 6 menunjukkan pengimplementasian dari algoritma IAM-PV pada tahap menghitung *total cost*.

```

1. int result=0;
2. for(i=0;i<jumlah_supply;i++)
3. {
4.     for(j=0;j<jumlah_demand;j++)
5.         result+=(allocation[i][j]*cost[i][j]);
6. }

```

Kode Sumber 6 Implementasi tahap menghitung Total Cost

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas uji coba dan evaluasi terhadap algoritma IAM-PV yang telah dikembangkan untuk mencari solusi dalam permasalahan *Transportation Problem*.

5.1 Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan tugas ini dilakukan pada lingkungan dan alat kakas sebagai berikut:

Prosesor	: Prosesor: Intel® Core™ i7-7500U CPU @ 2.70GHz (4 CPUs) ~2.9GHz
RAM	: 8192 MB
Jenis <i>Device</i>	: Laptop
Sistem Operasi	: Microsoft Windows 7 64-bit

5.2 Data Uji Coba

Data yang digunakan untuk uji coba algoritma IAM modifikasi adalah 40 *Problem Set Transportation Problem* yang terdiri atas 33 *Problem Set* dari berbagai paper dan 7 *Problem Set* yang di-generate secara random.

5.3 Hasil Uji Coba

Berikut ini adalah hasil uji coba pada 40 *Problem Set Transportation Problem*.

1. Uji Coba Soal 1

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Srinivasan & Thompson, 1977. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set* 5.1. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}
[C_{ij}] 3 \times 4 &= [3, 6, 3, 4]; [6, 5, 11, 15]; [1, 3, 10, 5] \\
[S_i] 3 \times 1 &= [80, 90, 55] \\
[D_j] 1 \times 4 &= [70, 60, 35, 60]
\end{aligned}$$

Problem Set 5.1 Uji Coba Soal *Transportation Problem*

Tabel. 25 Hasil tiap algoritma pada Problem Set 5.1

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
880	1255	1075	955	1075	1075	880

2. Uji Coba Soal 2

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Sen, Som, & Sinha, 2010. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.2*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma:

$$\begin{aligned}
[C_{ij}] 5 \times 5 &= [60, 120, 75, 180]; [58, 100, 60, 165]; [62, 110, 65, 170]; [65, 115, 80, 175]; [70, 135, 85, 195] \\
[S_i] 5 \times 1 &= [8000, 9200, 6250, 4900, 6100] \\
[D_j] 1 \times 5 &= [5000, 2000, 10000, 6000]
\end{aligned}$$

Problem Set 5.2 Uji Coba Soal *Transportation Problem*

Tabel. 26 Hasil tiap algoritma pada Problem Set 5.2

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
2146750	2174000	2383250	2164000	2210000	2210000	2158500

3. Uji Coba Soal 3

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Goyal, 1984. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.3*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}[C_{ij}] \ 3 \times 3 &= [6, 10, 14]; [12, 19, 21]; [15, 14, 17] \\ [S_i] \ 3 \times 1 &= [50, 50, 50] \\ [D_j] \ 1 \times 4 &= [30, 40, 55]\end{aligned}$$

Problem Set 5.3 Uji Coba Soal *Transportation Problem*

Tabel.27 Hasil tiap algoritma pada Problem Set 5.3

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
1650	1815	1885	1745	1695	1695	1650

4. Uji Coba Soal 4

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Deshmukh, 2012. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.4*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma:

$$\begin{aligned}[C_{ij}] \ 3 \times 4 &= [19, 30, 50, 10]; [70, 30, 40, 60]; [40, 8, 70, 20] \\ [S_i] \ 3 \times 1 &= [7, 9, 18] \\ [D_j] \ 1 \times 4 &= [5, 8, 7, 14]\end{aligned}$$

Problem Set 5.4 Uji Coba Soal *Transportation Problem*

Tabel.28 Hasil tiap algoritma pada Problem Set 5.4

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
743	1015	814	779	781	781	779

5. Uji Coba Soal 5

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Ramadan & Ramadan, 2012. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.5*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma:

$$\begin{aligned}[C_{ij}] \ 3 \times 3 &= [32, 40, 120]; [60, 68, 104]; [200, 80, 60] \\ [S_i] \ 3 \times 1 &= [20, 30, 45] \\ [D_j] \ 1 \times 3 &= [30, 35, 30]\end{aligned}$$

Problem Set 5.5 Uji Coba Soal *Transportation Problem*

Tabel.29 Hasil tiap algoritma pada Problem Set 5.5

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
5600	5600	5600	5600	5600	5600	5600

6. Uji Coba Soal 6

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Kulkarni, 2010. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.6*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}[C_{ij}] \ 4 \times 3 &= [3, 4, 6]; [7, 3, 8]; [6, 4, 5]; [7, 5, 2] \\ [S_i] \ 3 \times 1 &= [100, 80, 90, 120] \\ [D_j] \ 1 \times 4 &= [110, 110, 60]\end{aligned}$$

Problem Set 5. Uji Coba Soal *Transportation Problem*

Tabel.30 Hasil tiap algoritma pada Problem Set 5.6

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
840	1010	1210	880	1020	930	980

7. Uji Coba Soal 7

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Schrenk, Finke, & Cung, 2011. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.7*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 3 \times 4 &= [3, 6, 1, 5]; [7, 9, 2, 7]; [2, 4, 2, 1] \\ [S_i] \ 3 \times 1 &= [6, 6, 6] \\ [D_j] \ 1 \times 4 &= [4, 5, 4, 5] \end{aligned}$$

Problem Set 5.6 Uji Coba Soal *Transportation Problem*

Tabel.31 Hasil tiap algoritma pada Problem Set 5.7

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
59	64	69	59	66	69	69

8. Uji Coba Soal 8

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Samuel, 2012. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.8*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 3 \times 4 &= [1, 2, 3, 4]; [4, 3, 2, 0]; [0, 2, 2, 1] \\ [S_i] \ 3 \times 1 &= [6, 8, 10] \\ [D_j] \ 1 \times 4 &= [4, 6, 8, 6] \end{aligned}$$

Problem Set 5.7 Uji Coba Soal *Transportation Problem*

Tabel.32 Hasil tiap algoritma pada Problem Set 5.8

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
28	42	28	28	28	34	28

9. Uji Coba Soal 9

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Imam, Elsharawy, Gomah, & Samy, 2009. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set* 5.9. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 3 \times 4 &= [10, 2, 20, 11]; [12, 7, 9, 20]; [4, 14, 16, 18] \\ [S_i] \ 3 \times 1 &= [15, 25, 10] \\ [D_j] \ 1 \times 4 &= [5, 15, 15, 15] \end{aligned}$$

Problem Set 5.9 Uji Coba Soal *Transportation Problem*

Tabel.33 Hasil tiap algoritma pada Problem Set 5.9

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
435	520	475	475	520	520	520

10. Uji Coba Soal 10

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Adlakha & Kowalski, 2009. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set* 5.10. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 4 \times 5 &= [2, 1, 3, 2, 2]; [3, 2, 1, 1, 1]; [5, 4, 2, 1, 3]; [7, 5, 5, 3, 1] \\ [S_i] \ 4 \times 1 &= [20, 70, 30, 60] \\ [D_j] \ 1 \times 5 &= [50, 30, 30, 50, 20] \end{aligned}$$

Problem Set 5.10 Uji Coba Soal *Transportation Problem*

Tabel.34 Hasil tiap algoritma pada Problem Set 5.10

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
390	390	490	390	550	510	390

11. Uji Coba Soal 11

Permasalahan *Transportation Problem* pada *Problem-Set* 5.11 ini diambil dari sumber Juman & Hoque, 2015. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 4 \times 5 &= [25, 14, 34, 46, 45]; [10, 47, 14, 20, 41]; \\ &\quad [22, 42, 38, 21, 46]; [36, 20, 41, 38, 44] \\ [S_i] \ 3 \times 1 &= [27, 35, 37, 45] \\ [D_j] \ 1 \times 4 &= [22, 27, 28, 33, 34] \end{aligned}$$

Problem Set 5.11 Uji Coba Soal *Transportation Problem*

Tabel.35 Hasil tiap algoritma pada Problem Set 5.11

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
3458	4782	3572	3663	4147	3572	3513

12. Uji Coba Soal 12

Permasalahan *Transportation Problem* pada *Problem-Set* 5.12 ini diambil dari sumber Juman & Hoque, 2015. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 4 \times 6 &= [9, 12, 9, 6, 9, 10]; [7, 3, 7, 7, 5, 5]; [6, 5, 9, 11, 3, 11]; \\ &\quad [6, 8, 11, 2, 2, 10] \\ [S_i] \ 3 \times 1 &= [2, 5, 6, 9] \\ [D_j] \ 1 \times 4 &= [2, 2, 4, 4, 4, 6] \end{aligned}$$

Problem Set 5.12 Uji Coba Soal *Transportation Problem*

Tabel.36 Hasil tiap algoritma pada Problem Set 5.12

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
109	169	117	109	112	116	119

13. Uji Coba Soal 13

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Ahmed, Khan, Uddin, & Ahmed, 2016. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.13*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 4 \times 4 &= [7,5,9,11]; [4,3,8,6]; [3,8,10,5]; [2,6,7,3] \\ [S_i] \ 3 \times 1 &= [30,25,20,15] \\ [D_j] \ 1 \times 4 &= [30,30,20,10] \end{aligned}$$

Problem Set 5.13 Uji Coba Soal *Transportation Problem*

Tabel.37 Hasil tiap algoritma pada Problem Set 5.13

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
410	540	435	470	420	435	415

14. Uji Coba Soal 14

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Ahmed, Khan, Uddin, & Ahmed, 2016. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.14*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 3 \times 4 &= [3,1,7,4]; [2,6,5,9]; [8,3,3,2] \\ [S_i] \ 3 \times 1 &= [300,400,500] \\ [D_j] \ 1 \times 4 &= [250,350,400,200] \end{aligned}$$

Problem Set 5.14 Uji Coba Soal *Transportation Problem*

Tabel.38 Hasil tiap algoritma pada Problem Set 5.14

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
2850	4400	2850	2850	2850	2850	2850

15. Uji Coba Soal 15

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Ahmed, Khan, Uddin, & Ahmed, 2016. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.15*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 3 \times 5 &= [5, 7, 10, 5, 3]; [8, 6, 9, 12, 14]; [10, 9, 8, 10, 15] \\ [S_i] \ 3 \times 1 &= [5, 10, 10] \\ [D_j] \ 1 \times 5 &= [3, 3, 10, 5, 4] \end{aligned}$$

Problem Set 5.15 Uji Coba Soal *Transportation Problem*

Tabel.39 Hasil tiap algoritma pada Problem Set 5.15

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
183	234	191	187	183	183	183

16. Uji Coba Soal 16

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Uddin & Khan, 2016. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.16*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 3 \times 4 &= [19, 30, 50, 12]; [70, 30, 40, 60]; [40, 10, 60, 20] \\ [S_i] \ 3 \times 1 &= [7, 10, 18] \\ [D_j] \ 1 \times 4 &= [5, 8, 7, 15] \end{aligned}$$

Problem Set 5.16. Uji Coba Soal *Transportation Problem*

Tabel.40 Hasil tiap algoritma pada Problem Set 5.16

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
799	975	894	859	868	868	859

17. Uji Coba Soal 17

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Uddin & Khan, 2016. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.17*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}[C_{ij}] \ 3 \times 5 &= [4,1,2,4,4]; [2,3,2,2,2]; [3,5,2,4,4] \\ [S_i] \ 3 \times 1 &= [60,35,40] \\ [D_j] \ 1 \times 5 &= [22,45,20,18]\end{aligned}$$

Problem Set 5.17 Uji Coba Soal *Transportation Problem*

Tabel.41 Hasil tiap algoritma pada Problem Set 5.17

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
273	363	305	273	273	278	273

18. Uji Coba Soal 18

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Uddin & Khan, 2016. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.18*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}[C_{ij}] \ 3 \times 4 &= [6,1,9,3]; [11,5,2,8]; [10,12,4,7] \\ [S_i] \ 3 \times 1 &= [70,55,90] \\ [D_j] \ 1 \times 4 &= [85,35,50,45]\end{aligned}$$

Problem Set 5.18 Uji Coba Soal *Transportation Problem*

Tabel.42 Hasil tiap algoritma pada Problem Set 5.18

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
1160	1265	1165	1220	1315	1165	1160

19. Uji Coba Soal 19

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Khan, Vilcu, Sultana, & Ahmed, 2015. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.19*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}[C_{ij}] \ 3 \times 4 &= [3,6,8,4]; [6,1,2,5]; [7,8,3,9] \\ [S_i] \ 3 \times 1 &= [20,28,17] \\ [D_j] \ 1 \times 4 &= [15,19,13,18]\end{aligned}$$

Problem Set 5.19 Uji Coba Soal *Transportation Problem*

Tabel.43 Hasil tiap algoritma pada Problem Set 5.19

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
200	273	231	204	218	231	200

20. Uji Coba Soal 20

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Azad & Hossain, 2017. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.20*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}[C_{ij}] \ 3 \times 4 &= [9,8,5,7]; [4,6,8,7]; [5,8,9,5] \\ [S_i] \ 3 \times 1 &= [12,14,16] \\ [D_j] \ 1 \times 4 &= [8,18,13,3]\end{aligned}$$

Problem Set 5.20 Uji Coba Soal *Transportation Problem*

Tabel.44 Hasil tiap algoritma pada Problem Set 5.20

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
240	320	248	248	248	248	249

21. Uji Coba Soal 21

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Morade, 2017. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.21*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 3 \times 3 &= [50, 30, 220]; [90, 45, 170]; [250, 200, 50] \\ [S_i] \ 3 \times 1 &= [1, 3, 4] \\ [D_j] \ 1 \times 3 &= [4, 2, 2] \end{aligned}$$

Problem Set 5.21 Uji Coba Soal *Transportation Problem*

Tabel.45 Hasil tiap algoritma pada Problem Set 5.21

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
820	820	855	820	855	855	820

22. Uji Coba Soal 22

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Jude, 2016. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.22*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 3 \times 4 &= [5, 7, 9, 6]; [6, 7, 10, 5]; [7, 6, 8, 1] \\ [S_i] \ 3 \times 1 &= [12, 14, 10] \\ [D_j] \ 1 \times 4 &= [10, 6, 8, 12] \end{aligned}$$

Problem Set 5.22 Uji Coba Soal *Transportation Problem*

Tabel.46 Hasil tiap algoritma pada Problem Set 5.22

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
190	192	192	190	194	192	190

23. Uji Coba Soal 23

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Jude, 2016. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.23*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 4 \times 4 &= [2,5,6,3]; [9,6,2,1]; [5,2,3,6]; [7,7,2,4] \\ [S_i] \ 4 \times 1 &= [6,9,7,12] \\ [D_j] \ 1 \times 4 &= [10,4,6,14] \end{aligned}$$

Problem Set 5.23 Uji Coba Soal *Transportation Problem*

Tabel.47 Hasil tiap algoritma pada Problem Set 5.23

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
83	149	83	92	115	103	88

24. Uji Coba Soal 24

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Hosseini, 2017. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.24*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 3 \times 4 &= [20,22,17,4]; [24,37,9,7]; [32,37,20,15] \\ [S_i] \ 3 \times 1 &= [120,70,50] \\ [D_j] \ 1 \times 4 &= [60,40,30,110] \end{aligned}$$

Problem Set 5.24 Uji Coba Soal *Transportation Problem*

Tabel.48 Hasil tiap algoritma pada Problem Set 5.24

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
3460	3680	3670	3520	3790	3790	3570

25. Uji Coba Soal 25

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Hosseini, 2017. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.25*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}[C_{ij}] \ 3 \times 4 &= [3,5,7,6]; [2,5,8,2]; [3,6,9,2] \\ [S_i] \ 3 \times 1 &= [50,75,25] \\ [D_j] \ 1 \times 4 &= [20,20,50,60]\end{aligned}$$

Problem Set 5.25 Uji Coba Soal *Transportation Problem*

Tabel.49 Hasil tiap algoritma pada Problem Set 5.25

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
610	670	650	650	665	665	610

26. Uji Coba Soal 26

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Ahmed, Khan, Ahmed, & Uddin, 2016. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.26*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}[C_{ij}] \ 3 \times 3 &= [4,3,5]; [6,5,4]; [8,10,7] \\ [S_i] \ 3 \times 1 &= [90,80,100] \\ [D_j] \ 1 \times 3 &= [70,120,80]\end{aligned}$$

Problem Set 5.26 Uji Coba Soal *Transportation Problem*

Tabel.50 Hasil tiap algoritma pada Problem Set 5.26

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
1390	1500	1450	1500	1390	1390	1390

27. Uji Coba Soal 27

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.27*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}[C_{ij}] \ 3 \times 4 &= [20, 2, 20, 11]; [24, 7, 9, 20]; [8, 14, 16, 18] \\ [S_i] \ 3 \times 1 &= [30, 50, 20] \\ [D_j] \ 1 \times 4 &= [10, 30, 30, 30]\end{aligned}$$

Problem Set 5.27 Uji Coba Soal *Transportation Problem*

Tabel.51 Hasil tiap algoritma pada Problem Set 5.27

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
910	1140	990	990	1080	1080	990

28. Uji Coba Soal 28

Permasalahan *Transportation Problem* pada soal ini di-*generate* secara random. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.28*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}[C_{ij}] \ 4 \times 4 &= [7, 5, 27, 22]; [4, 3, 24, 12]; [6, 16, 60, 20]; [2, 6, 21, 6] \\ [S_i] \ 4 \times 1 &= [60, 50, 40, 30] \\ [D_j] \ 1 \times 4 &= [60, 60, 40, 20]\end{aligned}$$

Problem Set 5.28 Uji Coba Soal *Transportation Problem*

Tabel.52 Hasil tiap algoritma pada Problem Set 5.28

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
1670	2860	1940	1680	3280	2580	1690

29. Uji Coba Soal 29

Permasalahan *Transportation Problem* pada soal ini di-*generate* secara random. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.29*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$[C_{ij}] \ 4 \times 4$	$= [21,5,9,11]; [12,3,8,6]; [9,8,10,5]; [6,6,7,3]$
$[S_i] \ 4 \times 1$	$= [120,100,80,60]$
$[D_j] \ 1 \times 4$	$= [120,120,80,40]$

Problem Set 5.29 Uji Coba Soal *Transportation Problem*

Tabel.53 Hasil tiap algoritma pada Problem Set 5.29

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
2280	3840	2500	2400	2360	2500	2400

30. Uji Coba Soal 30

Permasalahan *Transportation Problem* pada soal ini di-*generate* secara random. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.30*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$[C_{ij}] \ 3 \times 4$	$= [4,6,9,5]; [2,6,4,1]; [5,7,2,9]$
$[S_i] \ 3 \times 1$	$= [16,12,15]$
$[D_j] \ 1 \times 4$	$= [12, 14, 9, 8]$

Problem Set 5.30 Uji Coba Soal *Transportation Problem*

Tabel.54 Hasil tiap algoritma pada Problem Set 5.40

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
156	226	156	156	156	156	156

31. Uji Coba Soal 31

Permasalahan *Transportation Problem* pada soal ini di-*generate* secara random. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.31*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}[C_{ij}] \ 3 \times 3 &= [7, 8, 7]; [18, 812]; [8, 12, 12] \\ [S_i] \ 3 \times 1 &= [13, 14, 8] \\ [D_j] \ 1 \times 3 &= [14, 8, 13]\end{aligned}$$

Problem Set 5.31 Uji Coba Soal *Transportation Problem*

Tabel.55 Hasil tiap algoritma pada Problem Set 5.31

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
291	329	319	327	319	319	291

32. Uji Coba Soal 32

Permasalahan *Transportation Problem* pada soal ini di-*generate* secara random. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.32*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}[C_{ij}] \ 3 \times 4 &= [1, 6, 5, 9]; [3, 2, 4, 15]; [8, 9, 3, 10] \\ [S_i] \ 3 \times 1 &= [20, 25, 15] \\ [D_j] \ 1 \times 4 &= [15, 15, 10, 20]\end{aligned}$$

Problem Set 5.32 Uji Coba Soal *Transportation Problem*

Tabel.56 Hasil tiap algoritma pada Problem Set 5.32

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
280	330	320	320	455	315	280

33. Uji Coba Soal 33

Permasalahan *Transportation Problem* pada soal ini di-generate secara random. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.33*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma

$$\begin{aligned}[C_{ij}] \ 3 \times 3 &= [6,1,9]; [11,5,2]; [10,12,14] \\ [S_i] \ 3 \times 1 &= [70,50,50] \\ [D_j] \ 1 \times 3 &= [85,35,50]\end{aligned}$$

Problem Set 5.33 Uji Coba Soal *Transportation Problem*

Tabel.57 Hasil tiap algoritma pada Problem Set 5.33

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
845	960	845	960	845	845	845

34. Uji Coba Soal 34

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Amirul Islam, 2012. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.34*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}[C_{ij}] \ 3 \times 3 &= [4,8,11]; [6,3,8]; [7,6,5] \\ [S_i] \ 3 \times 1 &= [12,10,9] \\ [D_j] \ 1 \times 3 &= [13,8,10]\end{aligned}$$

Problem Set 5.34 Uji Coba Soal *Transportation Problem*

Tabel.58 Hasil tiap algoritma pada Problem Set 5.34

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
131	131	131	131	136	136	131

35. Uji Coba Soal 35

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Mondal, 2012. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.35*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 3 \times 4 &= [12, 4, 9, 5]; [8, 1, 6, 6]; [1, 2, 4, 7] \\ [S_i] \ 3 \times 1 &= [55, 40, 30] \\ [D_j] \ 1 \times 4 &= [40, 20, 45, 20] \end{aligned}$$

Problem Set 5.35 Uji Coba Soal *Transportation Problem*

Tabel.59 Hasil tiap algoritma pada Problem Set 5.35

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
605	935	615	605	605	605	605

36. Uji Coba Soal 36

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Kanti Das, 2014. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.36*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 3 \times 4 &= [6, 8, 10, 9]; [5, 8, 11, 5]; [6, 9, 12, 5] \\ [S_i] \ 3 \times 1 &= [50, 75, 25] \\ [D_j] \ 1 \times 4 &= [20, 20, 50, 60] \end{aligned}$$

Problem Set 5.36 Uji Coba Soal *Transportation Problem*

Tabel.60 Hasil tiap algoritma pada Problem Set 5.36

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
1060	1120	1100	1100	1115	1115	1060

37. Uji Coba Soal 37

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Kanti Das, 2014. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.37*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}[C_{ij}] \ 5 \times 4 &= [10, 20, 5, 7]; [13, 9, 13, 8]; [4, 15, 7, 9]; [14, 7, 1, 1]; \\ &\quad [3, 12, 5, 19] \\ [S_i] \ 5 \times 1 &= [200, 300, 200, 400, 400] \\ [D_j] \ 1 \times 4 &= [500, 600, 200, 200]\end{aligned}$$

Problem Set 5.37 Uji Coba Soal *Transportation Problem*

Tabel.61 Hasil tiap algoritma pada Problem Set 5.37

Optimal	NWC M	LCM	VA M	IAM	IAM- IL	IAM- PV
8200	16500	10200	9800	10200	10200	8200

38. Uji Coba Soal 38

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Molah Mesbahudin, 2015. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.38*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned}[C_{ij}] \ 3 \times 3 &= [15, 7, 25]; [8, 12, 14]; [17, 19, 21] \\ [S_i] \ 3 \times 1 &= [12, 17, 7] \\ [D_j] \ 1 \times 3 &= [12, 10, 14]\end{aligned}$$

Problem Set 5.38 Uji Coba Soal *Transportation Problem*

Tabel.62 Hasil tiap algoritma pada Problem Set 5.38

Optimal	NWCM	LCM	VAM	IAM	IAM- IL	IAM- PV
425	545	433	425	425	425	425

39. Uji Coba Soal 39

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Molah Mesbahudin, 2015. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.39*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

$$\begin{aligned} [C_{ij}] \ 3 \times 3 &= [13, 21, 14]; [8, 12, 21]; [15, 17, 19] \\ [S_i] \ 3 \times 1 &= [13, 20, 5] \\ [D_j] \ 1 \times 3 &= [12, 15, 11] \end{aligned}$$

Problem Set 5.39 Uji Coba Soal *Transportation Problem*

Tabel.63 Hasil tiap algoritma pada Problem Set 5.39

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
465	566	473	473	473	473	473

40. Uji Coba Soal 40

Permasalahan *Transportation Problem* pada soal ini diambil dari sumber Molah Mesbahudin, 2015. Diketahui notasi sebuah transportation problem seperti pada *Problem-Set 5.40*. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma

$$\begin{aligned} [C_{ij}] \ 3 \times 3 &= [4, 3, 5]; [6, 5, 4]; [8, 10, 7] \\ [S_i] \ 3 \times 1 &= [9, 8, 10] \\ [D_j] \ 1 \times 3 &= [7, 12, 8] \end{aligned}$$

Problem Set 5.40 Uji Coba Soal *Transportation Problem*

Tabel.64 Hasil tiap algoritma pada Problem Set 5.40

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
139	150	145	150	139	139	139

41. Uji Coba Soal Khusus

Permasalahan *Transportation Problem* pada soal ini dibuat berdasarkan dimensi maksimal yang dapat diselesaikan dengan *tools* ToraSystem yaitu 100x100. *Problem Set Transportation Problem* 100x100 terdapat pada halaman lampiran. Berikut ini adalah hasil yang didapat dengan menggunakan berbagai algoritma :

Tabel.65 Hasil tiap algoritma pada Problem Set Khusus

Optimal	NWCM	LCM	VAM	IAM	IAM-IL	IAM-PV
100000	100000	1000 00	10000 0	1000 00	10000 0	1000 00

5.4 Evaluasi Algoritma IAM-PV

Pada tabel 66 berikut ini menunjukkan rekapitulasi tiap algoritma pada soal yang telah diujicobakan

Tabel.66 Nilai Akurasi Algoritma

Algoritma	Jumlah Optimal	Jumlah Tidak Optimal
NWCM	4	36
LCM	7	33
VAM	13	27
IAM	11	29
IAM-IL	9	31
IAM-PV	24	16
Tools ToraSystem	40	0

Pada sub-bab ini, akan diberikan perbandingan nilai-nilai algoritma IAM-PV terhadap algoritma lainnya. Nilai yang akan dihitung antara lain, Persentase Keberhasilan, Persentase Kegagalan, Rata-rata Kegagalan, Persentase Deviasi, dan Persentase Improvement.

Persentase Keberhasilan dihitung dari perbandingan jumlah benar solusi dari algoritma yang dimaksud berbanding dengan banyaknya soal, dimana dalam penelitian ini terdapat 40 soal. Yang dimaksud benar di sini adalah apabila solusi yang dihasilkan dari algoritma tersebut sama dengan hasil optimal yang didapat dari ToraSystem. Persentase Keberhasilan semakin besar semakin baik. Berikut ini adalah rumus Persentase Keberhasilan

$$\text{Persentase Keberhasilan} = \frac{\text{Jumlah Optimal}}{\text{Total Semua Soal}} \times 100\%$$

Persentase Kegagalan adalah komplemen dari Persentase Keberhasilan, yaitu jumlah salah berbanding banyaknya soal. Persentase Kegagalan semakin kecil semakin baik. Rumus Persentase Kegagalan adalah sebagai berikut

$$\text{Persentase Kegagalan} = \frac{\text{Jumlah Tidak Optimal}}{\text{Total Semua Soal}} \times 100\%$$

Rata-Rata Kegagalan adalah nilai rata-rata dari lebarnya *error* semua solusi yang bernilai salah. Lebar *error* didapat dari selisih solusi yang didapat dari algoritma yang dimaksud dengan hasil optimal yang didapat dari ToraSystem. Rata-Rata Kegagalan semakin kecil semakin baik. Berikut ini adalah rumus Rata-Rata Kegagalan.

$$\text{Rata-Rata Kegagalan} = \frac{\sum_{n=1}^j \text{Solusi IBFS} - \text{Solusi Optimal}}{\text{Jumlah Tidak Optimal}}$$

Persentase Deviasi dihitung dari Perbandingan antara lebar *error* dengan solusi optimal yang didapat dari ToraSystem. Persentase Deviasi semakin kecil semakin baik. Berikut ini adalah rumus Persentase Deviasi.

$$\text{Persentase Deviasi} = \frac{\text{Solusi IBFS} - \text{Solusi Optimal}}{\text{Solusi Optimal}} \times 100\%$$

Persentase Improvement adalah metode untuk membandingkan algoritma *Transportation Problem*. Pada Persentase Improvement terdapat 2 algoritma yang dibandingkan, algoritma 1 dan algoritma 2. Apabila nilai Persentase Improvement bernilai positif, maka dapat disimpulkan bahwa algoritma 1 lebih baik dari algoritma 2. Apabila Persentase Improvement bernilai negatif berarti sebaliknya. Berikut ini adalah rumus Persentase Improvement

$$\text{Persentase Improvement} = \frac{\text{Solusi IBFS1} - \text{Solusi IBFS2}}{\text{Solusi IBFS1}} \times 100\%$$

Berdasarkan tabel 67, algoritma IAM-PV memiliki presentase benar tertinggi yaitu 60%. Apabila dibandingkan dengan algoritma IAM, kenaikan tingkat akurasi algoritma IAM-PV mencapai 32,5%. Selain itu, algoritma IAM-PV juga memiliki rata-rata kegagalan terendah yaitu 346,306. Nilai rata-rata kegagalan ini menurun sebesar 1583,22 dari algoritma IAM. Pada tabel 68 berikut ini menunjukkan nilai *improvement* algoritma IAM-PV terhadap algoritma IAM.

Tabel.67 Nilai Persentase Berhasil dan Gagal

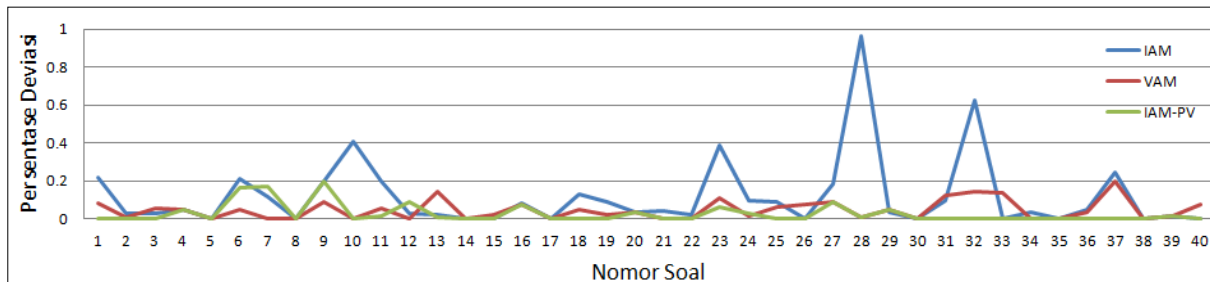
Algoritma	Persentase Berhasil	Persentase Gagal	Rata-rata Gagal
NWCM	10%	90%	1239,1
LCM	17,5%	82,%	6691,8
VAM	32,5%	67,5%	561,556
IAM	27,5%	72,5%	1930,53
IAM-IL	22,5%	77,5%	1888,22
IAM-PV	60%	40%	347,306

Tabel.68 Persentase Deviasi dan *Improvement* IAM pada IAM-PV

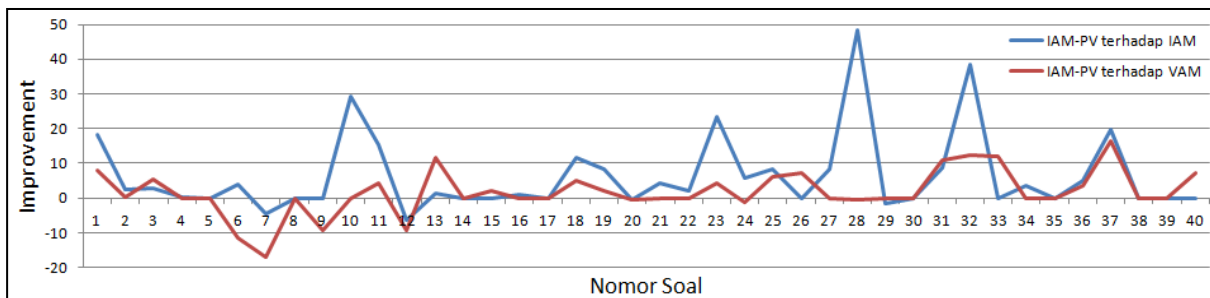
No	Hasil				Persentase Deviasi			Improvement	
	Optimal	IAM	VAM	IAM-PV	IAM	VAM	IAM-PV	IAM-PV terhadap VAM	IAM-PV terhadap IAM
1	880	1075	955	880	0.221	0.085	0	18.139	7.853
2	2146750	2210000	2164000	2158500	0.029	0.008	0.005	2.330	0.254
3	1650	1695	1745	1650	0.027	0.057	0	2.654	5.444
4	743	781	779	779	0.051	0.048	0.0484	0.256	0
5	5600	5600	5600	5600	0	0	0	0	0
6	840	1020	880	980	0.214	0.047	0.166	3.921	-11.363
7	59	66	59	69	0.118	0	0.169	-4.545	-16.949
8	28	28	28	28	0	0	0	0	0
9	435	520	475	520	0.195	0.091	0.195	0	-9.473
10	390	550	390	390	0.410	0	0	29.090	0
11	3458	4147	3663	3513	0.199	0.059	0.015	15.288	4.095
12	109	112	109	119	0.027	0	0.091	-6.25	-9.174

13	410	420	470	415	0.024	0.146	0.012	1.190	11.702
14	2850	2850	2850	2850	0	0	0	0	0
15	183	183	187	183	0	0.021	0	0	2.139
16	799	868	859	859	0.086	0.075	0.075	1.036	0
17	273	273	273	273	0	0	0	0	0
18	1160	1315	1220	1160	0.133	0.051	0	11.787	4.918
19	200	218	204	200	0.090	0.02	0	8.256	1.960
20	240	248	248	249	0.033	0.033	0.037	-0.403	-0.403
21	820	855	820	820	0.042	0	0	4.093	0
22	190	194	190	190	0.021	0	0	2.061	0
23	83	115	92	88	0.385	0.108	0.060	23.478	4.349
24	3460	3790	3520	3570	0.095	0.017	0.031	5.804	-1.420
25	610	665	650	610	0.090	0.065	0	8.270	6.154
26	1390	1390	1500	1390	0	0.079	0	0	7.333
27	910	1080	990	990	0.186	0.088	0.087	8.333	0
28	1670	3280	1680	1690	0.964	0.006	0.011	48.475	-0.595
29	2280	2360	2400	2400	0.035	0.053	0.052	-1.694	0

30	156	156	156	156	0	0	0	0	0
31	291	319	327	291	0.096	0.124	0	8.777	11.009
32	280	455	320	280	0.625	0.143	0	38.461	12.5
33	845	845	960	845	0	0.136	0	0	11.979
34	131	136	131	131	0.038	0	0	3.676	0
35	605	605	605	605	0	0	0	0	0
36	1060	1115	1100	1060	0.051	0.038	0	4.932	3.636
37	8200	10200	9800	8200	0.243	0.195	0	19.607	16.327
38	425	425	425	425	0	0	0	0	0
39	465	473	473	473	0.0172	0.017	0.017	0	0
40	139	139	150	139	0	0.079	0	0	7.333
Rata-rata					11.889	4.738	2.699	6.426	1.740



Gambar 8. Grafik Persentase deviasi



Gambar 9. Grafik Persentase Improvement

Berdasarkan tabel 68, nilai persentase deviasi algoritma IAM-PV adalah 2,699%, yang artinya 9,19% lebih baik dari algoritma IAM yang sebesar 11,889%. Selain itu, nilai *improvement* algoritma IAM-PV terhadap algoritma IAM bernilai positif yaitu 6,426%. Dengan demikian dapat disimpulkan bahwa algoritma IAM-PV dapat menghasilkan solusi yang lebih baik.

Grafik pada gambar 8 menunjukkan perbandingan Persentase Deviasi algoritma IAM, VAM, IAM-PV. Sedangkan grafik pada gambar 10 menunjukkan perbandingan Improvement algoritma IAM-PV terhadap IAM dan VAM.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan tugas akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang

6.1. Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian yang dilakukan, dapat diambil kesimpulan sebagai berikut.

1. Hasil uji coba pada penelitian ini menunjukkan bahwa algoritma IAM belum cukup optimal. Hal ini terbukti dengan nilai persentase keberhasilan yang cukup rendah yaitu 27,5%, serta nilai persentase kegagalan, rata-rata persentase deviasi, dan rata-rata kegagalan yang cukup tinggi masing-masing 72,5%, 11,889%, dan 1930,53.
2. Hasil uji coba pada penelitian ini menunjukkan bahwa algoritma IAM-PV mendekati optimal. Hal ini terbukti pada nilai persentase keberhasilan yang naik 32,5% dari 27,5% menjadi 60%. Selain itu, rata-rata kegagalan juga turun sebanyak 1583,22 dari 1930,53 menjadi 347,306. Rata-rata persentase deviasi juga turun sebesar 9,19% dari 11,889% menjadi 2,699%.
3. Nilai *improvement* pada algoritma IAM-PV terhadap algoritma IAM bernilai positif, sehingga hal ini membuktikan bahwa algoritma IAM-PV lebih baik daripada algoritma IAM

6.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada

hasil perancangan, implementasi, dan pengujian yang telah dilakukan.

1. Algoritma IAM-PV mempunyai nilai error yang cukup rendah, namun pada beberapa *problem set* masih ditemui kegagalan. sehingga harapannya di masa depan algoritma ini akan dikembangkan pada tahap-tahp tertentu
2. Pengaplikasian algoritma IBFS dengan menggunakan Dev-C++ cukup memakan waktu karena beberapa fungsi tidak tersedia dalam *library* seperti fungsi mencari nilai minmal, fungsi *men-sorting*, dan lain-lain. Disarankan untuk menggunakan *tools* lebih baik atau bahaasa pemrograman yang lebih modern.
3. Pada soal bertipe *Unbalance Problem*, program masih belum bisa menyesuaikan otomatis sehingga diperlukan penambahan *dummy*. Kedepannya diharapkan program yang dibuat dapat meng-*generate* sendiri penambahan *dummy* secara otomatis

DAFTAR PUSTAKA

- [1] Eghbal Hosseini, Three New Methods to Find Initial Basic Feasible Solution of Transportation Problems, *Applied Mathematical Sciences*, 11, (37) (2017), 1803-1814
- [2] Mollah Mesbahuddin Ahmed, Aminur Rahman Khan, Faruque Ahmed, Md. Sharif Uddin, Incessant Allocation Method for Solving Transportation Problems, *American Journal of Operations Research*, 6, (2016), 236-244
- [3] Mollah Mesbahuddin Ahmed, Aminur Rahman Khan, Md. Sharif Uddin, Faruque Ahmed, A New Approach to Solve Transportation Problems, *Open Journal of Optimization*, 5, (2016), 22-30
- [4] Z.A.M.S. Juman, M.A. Hoque, An efficient heuristic to obtain a better initial feasible solution to the transportation problem, *Applied Soft Computing*, 34, (2015), 813–826
- [5] Md Sharif Uddin¹, Aminur Rahman Khan¹, Chowdhury Golam Kibria, Iliyana Raeva, Improved Least Cost Method to Obtain a Better IBFS to the Transportation Problem, *Journal of Applied Mathematics & Bioinformatics*, 6, (2) (2016), 1-20

[Halaman ini sengaja dikosongkan]

LAMPIRAN

Uji Coba Transportation Problem dengan dimensi
100x100. Berikut ini adalah *Problem Set* yang diberikan

[C_{ij}] 100x100 =

1	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	30	31	32	33	34	35	36
	37	38	39	40	41	42	43
	44	45	46	47	48	49	50
	51	52	53	54	55	56	57
	58	59	60	61	62	63	64
	65	66	67	68	69	70	71
	72	73	74	75	76	77	78
	79	80	81	82	83	84	85
	86	87	88	89	90	91	92
	93	94	95	96	97	98	99
	100						
2	3	4	5	6	7	8	9
	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
	24	25	26	27	28	29	30
	31	32	33	34	35	36	37
	38	39	40	41	42	43	44
	45	46	47	48	49	50	51
	52	53	54	55	56	57	58
	59	60	61	62	63	64	65
	66	67	68	69	70	71	72
	73	74	75	76	77	78	79
	80	81	82	83	84	85	86
	87	88	89	90	91	92	93

	94	95	96	97	98	99	100
	101						
3	4	5	6	7	8	9	10
	11	12	13	14	15	16	17
	18	19	20	21	22	23	24
	25	26	27	28	29	30	31
	32	33	34	35	36	37	38
	39	40	41	42	43	44	45
	46	47	48	49	50	51	52
	53	54	55	56	57	58	59
	60	61	62	63	64	65	66
	67	68	69	70	71	72	73
	74	75	76	77	78	79	80
	81	82	83	84	85	86	87
	88	89	90	91	92	93	94
	95	96	97	98	99	100	101
	102						
4	5	6	7	8	9	10	11
	12	13	14	15	16	17	18
	19	20	21	22	23	24	25
	26	27	28	29	30	31	32
	33	34	35	36	37	38	39
	40	41	42	43	44	45	46
	47	48	49	50	51	52	53
	54	55	56	57	58	59	60
	61	62	63	64	65	66	67
	68	69	70	71	72	73	74
	75	76	77	78	79	80	81
	82	83	84	85	86	87	88
	89	90	91	92	93	94	95
	96	97	98	99	100	101	102
	103						
5	6	7	8	9	10	11	12
	13	14	15	16	17	18	19

	20	21	22	23	24	25	26
	27	28	29	30	31	32	33
	34	35	36	37	38	39	40
	41	42	43	44	45	46	47
	48	49	50	51	52	53	54
	55	56	57	58	59	60	61
	62	63	64	65	66	67	68
	69	70	71	72	73	74	75
	76	77	78	79	80	81	82
	83	84	85	86	87	88	89
	90	91	92	93	94	95	96
	97	98	99	100	101	102	103
	104						
6	7	8	9	10	11	12	13
	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
	28	29	30	31	32	33	34
	35	36	37	38	39	40	41
	42	43	44	45	46	47	48
	49	50	51	52	53	54	55
	56	57	58	59	60	61	62
	63	64	65	66	67	68	69
	70	71	72	73	74	75	76
	77	78	79	80	81	82	83
	84	85	86	87	88	89	90
	91	92	93	94	95	96	97
	98	99	100	101	102	103	104
	105						
7	8	9	10	11	12	13	14
	15	16	17	18	19	20	21
	22	23	24	25	26	27	28
	29	30	31	32	33	34	35
	36	37	38	39	40	41	42
	43	44	45	46	47	48	49

	50	51	52	53	54	55	56
	57	58	59	60	61	62	63
	64	65	66	67	68	69	70
	71	72	73	74	75	76	77
	78	79	80	81	82	83	84
	85	86	87	88	89	90	91
	92	93	94	95	96	97	98
	99	100	101	102	103	104	105
	106						
8	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	30	31	32	33	34	35	36
	37	38	39	40	41	42	43
	44	45	46	47	48	49	50
	51	52	53	54	55	56	57
	58	59	60	61	62	63	64
	65	66	67	68	69	70	71
	72	73	74	75	76	77	78
	79	80	81	82	83	84	85
	86	87	88	89	90	91	92
	93	94	95	96	97	98	99
	100	101	102	103	104	105	106
	107						
9	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
	24	25	26	27	28	29	30
	31	32	33	34	35	36	37
	38	39	40	41	42	43	44
	45	46	47	48	49	50	51
	52	53	54	55	56	57	58
	59	60	61	62	63	64	65
	66	67	68	69	70	71	72
	73	74	75	76	77	78	79

	80	81	82	83	84	85	86
	87	88	89	90	91	92	93
	94	95	96	97	98	99	100
	101	102	103	104	105	106	107
	108						
10	11	12	13	14	15	16	17
	18	19	20	21	22	23	24
	25	26	27	28	29	30	31
	32	33	34	35	36	37	38
	39	40	41	42	43	44	45
	46	47	48	49	50	51	52
	53	54	55	56	57	58	59
	60	61	62	63	64	65	66
	67	68	69	70	71	72	73
	74	75	76	77	78	79	80
	81	82	83	84	85	86	87
	88	89	90	91	92	93	94
	95	96	97	98	99	100	101
	102	103	104	105	106	107	108
	109						
11	12	13	14	15	16	17	18
	19	20	21	22	23	24	25
	26	27	28	29	30	31	32
	33	34	35	36	37	38	39
	40	41	42	43	44	45	46
	47	48	49	50	51	52	53
	54	55	56	57	58	59	60
	61	62	63	64	65	66	67
	68	69	70	71	72	73	74
	75	76	77	78	79	80	81
	82	83	84	85	86	87	88
	89	90	91	92	93	94	95
	96	97	98	99	100	101	102

	103	104	105	106	107	108	109
	110						
12	13	14	15	16	17	18	19
	20	21	22	23	24	25	26
	27	28	29	30	31	32	33
	34	35	36	37	38	39	40
	41	42	43	44	45	46	47
	48	49	50	51	52	53	54
	55	56	57	58	59	60	61
	62	63	64	65	66	67	68
	69	70	71	72	73	74	75
	76	77	78	79	80	81	82
	83	84	85	86	87	88	89
	90	91	92	93	94	95	96
	97	98	99	100	101	102	103
	104	105	106	107	108	109	110
	111						
13	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
	28	29	30	31	32	33	34
	35	36	37	38	39	40	41
	42	43	44	45	46	47	48
	49	50	51	52	53	54	55
	56	57	58	59	60	61	62
	63	64	65	66	67	68	69
	70	71	72	73	74	75	76
	77	78	79	80	81	82	83
	84	85	86	87	88	89	90
	91	92	93	94	95	96	97
	98	99	100	101	102	103	104
	105	106	107	108	109	110	111
	112						
14	15	16	17	18	19	20	21
	22	23	24	25	26	27	28

	29	30	31	32	33	34	35
	36	37	38	39	40	41	42
	43	44	45	46	47	48	49
	50	51	52	53	54	55	56
	57	58	59	60	61	62	63
	64	65	66	67	68	69	70
	71	72	73	74	75	76	77
	78	79	80	81	82	83	84
	85	86	87	88	89	90	91
	92	93	94	95	96	97	98
	99	100	101	102	103	104	105
	106	107	108	109	110	111	112
	113						
15	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	30	31	32	33	34	35	36
	37	38	39	40	41	42	43
	44	45	46	47	48	49	50
	51	52	53	54	55	56	57
	58	59	60	61	62	63	64
	65	66	67	68	69	70	71
	72	73	74	75	76	77	78
	79	80	81	82	83	84	85
	86	87	88	89	90	91	92
	93	94	95	96	97	98	99
	100	101	102	103	104	105	106
	107	108	109	110	111	112	113
	114						
16	17	18	19	20	21	22	23
	24	25	26	27	28	29	30
	31	32	33	34	35	36	37
	38	39	40	41	42	43	44
	45	46	47	48	49	50	51
	52	53	54	55	56	57	58

	59	60	61	62	63	64	65
	66	67	68	69	70	71	72
	73	74	75	76	77	78	79
	80	81	82	83	84	85	86
	87	88	89	90	91	92	93
	94	95	96	97	98	99	100
	101	102	103	104	105	106	107
	108	109	110	111	112	113	114
	115						
17	18	19	20	21	22	23	24
	25	26	27	28	29	30	31
	32	33	34	35	36	37	38
	39	40	41	42	43	44	45
	46	47	48	49	50	51	52
	53	54	55	56	57	58	59
	60	61	62	63	64	65	66
	67	68	69	70	71	72	73
	74	75	76	77	78	79	80
	81	82	83	84	85	86	87
	88	89	90	91	92	93	94
	95	96	97	98	99	100	101
	102	103	104	105	106	107	108
	109	110	111	112	113	114	115
	116						
18	19	20	21	22	23	24	25
	26	27	28	29	30	31	32
	33	34	35	36	37	38	39
	40	41	42	43	44	45	46
	47	48	49	50	51	52	53
	54	55	56	57	58	59	60
	61	62	63	64	65	66	67
	68	69	70	71	72	73	74
	75	76	77	78	79	80	81
	82	83	84	85	86	87	88

	89	90	91	92	93	94	95
	96	97	98	99	100	101	102
	103	104	105	106	107	108	109
	110	111	112	113	114	115	116
	117						
19	20	21	22	23	24	25	26
	27	28	29	30	31	32	33
	34	35	36	37	38	39	40
	41	42	43	44	45	46	47
	48	49	50	51	52	53	54
	55	56	57	58	59	60	61
	62	63	64	65	66	67	68
	69	70	71	72	73	74	75
	76	77	78	79	80	81	82
	83	84	85	86	87	88	89
	90	91	92	93	94	95	96
	97	98	99	100	101	102	103
	104	105	106	107	108	109	110
	111	112	113	114	115	116	117
	118						
20	21	22	23	24	25	26	27
	28	29	30	31	32	33	34
	35	36	37	38	39	40	41
	42	43	44	45	46	47	48
	49	50	51	52	53	54	55
	56	57	58	59	60	61	62
	63	64	65	66	67	68	69
	70	71	72	73	74	75	76
	77	78	79	80	81	82	83
	84	85	86	87	88	89	90
	91	92	93	94	95	96	97
	98	99	100	101	102	103	104
	105	106	107	108	109	110	111

	112	113	114	115	116	117	118
	119						
21	22	23	24	25	26	27	28
	29	30	31	32	33	34	35
	36	37	38	39	40	41	42
	43	44	45	46	47	48	49
	50	51	52	53	54	55	56
	57	58	59	60	61	62	63
	64	65	66	67	68	69	70
	71	72	73	74	75	76	77
	78	79	80	81	82	83	84
	85	86	87	88	89	90	91
	92	93	94	95	96	97	98
	99	100	101	102	103	104	105
	106	107	108	109	110	111	112
	113	114	115	116	117	118	119
	120						
22	23	24	25	26	27	28	29
	30	31	32	33	34	35	36
	37	38	39	40	41	42	43
	44	45	46	47	48	49	50
	51	52	53	54	55	56	57
	58	59	60	61	62	63	64
	65	66	67	68	69	70	71
	72	73	74	75	76	77	78
	79	80	81	82	83	84	85
	86	87	88	89	90	91	92
	93	94	95	96	97	98	99
	100	101	102	103	104	105	106
	107	108	109	110	111	112	113
	114	115	116	117	118	119	120
	121						
23	24	25	26	27	28	29	30
	31	32	33	34	35	36	37

	38	39	40	41	42	43	44
	45	46	47	48	49	50	51
	52	53	54	55	56	57	58
	59	60	61	62	63	64	65
	66	67	68	69	70	71	72
	73	74	75	76	77	78	79
	80	81	82	83	84	85	86
	87	88	89	90	91	92	93
	94	95	96	97	98	99	100
	101	102	103	104	105	106	107
	108	109	110	111	112	113	114
	115	116	117	118	119	120	121
	122						
24	25	26	27	28	29	30	31
	32	33	34	35	36	37	38
	39	40	41	42	43	44	45
	46	47	48	49	50	51	52
	53	54	55	56	57	58	59
	60	61	62	63	64	65	66
	67	68	69	70	71	72	73
	74	75	76	77	78	79	80
	81	82	83	84	85	86	87
	88	89	90	91	92	93	94
	95	96	97	98	99	100	101
	102	103	104	105	106	107	108
	109	110	111	112	113	114	115
	116	117	118	119	120	121	122
	123						
25	26	27	28	29	30	31	32
	33	34	35	36	37	38	39
	40	41	42	43	44	45	46
	47	48	49	50	51	52	53
	54	55	56	57	58	59	60
	61	62	63	64	65	66	67

	68	69	70	71	72	73	74
	75	76	77	78	79	80	81
	82	83	84	85	86	87	88
	89	90	91	92	93	94	95
	96	97	98	99	100	101	102
	103	104	105	106	107	108	109
	110	111	112	113	114	115	116
	117	118	119	120	121	122	123
	124						
26	27	28	29	30	31	32	33
	34	35	36	37	38	39	40
	41	42	43	44	45	46	47
	48	49	50	51	52	53	54
	55	56	57	58	59	60	61
	62	63	64	65	66	67	68
	69	70	71	72	73	74	75
	76	77	78	79	80	81	82
	83	84	85	86	87	88	89
	90	91	92	93	94	95	96
	97	98	99	100	101	102	103
	104	105	106	107	108	109	110
	111	112	113	114	115	116	117
	118	119	120	121	122	123	124
	125						
27	28	29	30	31	32	33	34
	35	36	37	38	39	40	41
	42	43	44	45	46	47	48
	49	50	51	52	53	54	55
	56	57	58	59	60	61	62
	63	64	65	66	67	68	69
	70	71	72	73	74	75	76
	77	78	79	80	81	82	83
	84	85	86	87	88	89	90
	91	92	93	94	95	96	97

	98	99	100	101	102	103	104
	105	106	107	108	109	110	111
	112	113	114	115	116	117	118
	119	120	121	122	123	124	125
	126						
28	29	30	31	32	33	34	35
	36	37	38	39	40	41	42
	43	44	45	46	47	48	49
	50	51	52	53	54	55	56
	57	58	59	60	61	62	63
	64	65	66	67	68	69	70
	71	72	73	74	75	76	77
	78	79	80	81	82	83	84
	85	86	87	88	89	90	91
	92	93	94	95	96	97	98
	99	100	101	102	103	104	105
	106	107	108	109	110	111	112
	113	114	115	116	117	118	119
	120	121	122	123	124	125	126
	127						
29	30	31	32	33	34	35	36
	37	38	39	40	41	42	43
	44	45	46	47	48	49	50
	51	52	53	54	55	56	57
	58	59	60	61	62	63	64
	65	66	67	68	69	70	71
	72	73	74	75	76	77	78
	79	80	81	82	83	84	85
	86	87	88	89	90	91	92
	93	94	95	96	97	98	99
	100	101	102	103	104	105	106
	107	108	109	110	111	112	113
	114	115	116	117	118	119	120

	121	122	123	124	125	126	127
	128						
30	31	32	33	34	35	36	37
	38	39	40	41	42	43	44
	45	46	47	48	49	50	51
	52	53	54	55	56	57	58
	59	60	61	62	63	64	65
	66	67	68	69	70	71	72
	73	74	75	76	77	78	79
	80	81	82	83	84	85	86
	87	88	89	90	91	92	93
	94	95	96	97	98	99	100
	101	102	103	104	105	106	107
	108	109	110	111	112	113	114
	115	116	117	118	119	120	121
	122	123	124	125	126	127	128
	129						
31	32	33	34	35	36	37	38
	39	40	41	42	43	44	45
	46	47	48	49	50	51	52
	53	54	55	56	57	58	59
	60	61	62	63	64	65	66
	67	68	69	70	71	72	73
	74	75	76	77	78	79	80
	81	82	83	84	85	86	87
	88	89	90	91	92	93	94
	95	96	97	98	99	100	101
	102	103	104	105	106	107	108
	109	110	111	112	113	114	115
	116	117	118	119	120	121	122
	123	124	125	126	127	128	129
	130						
32	33	34	35	36	37	38	39
	40	41	42	43	44	45	46

	47	48	49	50	51	52	53
	54	55	56	57	58	59	60
	61	62	63	64	65	66	67
	68	69	70	71	72	73	74
	75	76	77	78	79	80	81
	82	83	84	85	86	87	88
	89	90	91	92	93	94	95
	96	97	98	99	100	101	102
	103	104	105	106	107	108	109
	110	111	112	113	114	115	116
	117	118	119	120	121	122	123
	124	125	126	127	128	129	130
	131						
33	34	35	36	37	38	39	40
	41	42	43	44	45	46	47
	48	49	50	51	52	53	54
	55	56	57	58	59	60	61
	62	63	64	65	66	67	68
	69	70	71	72	73	74	75
	76	77	78	79	80	81	82
	83	84	85	86	87	88	89
	90	91	92	93	94	95	96
	97	98	99	100	101	102	103
	104	105	106	107	108	109	110
	111	112	113	114	115	116	117
	118	119	120	121	122	123	124
	125	126	127	128	129	130	131
	132						
34	35	36	37	38	39	40	41
	42	43	44	45	46	47	48
	49	50	51	52	53	54	55
	56	57	58	59	60	61	62
	63	64	65	66	67	68	69
	70	71	72	73	74	75	76

	77	78	79	80	81	82	83
	84	85	86	87	88	89	90
	91	92	93	94	95	96	97
	98	99	100	101	102	103	104
	105	106	107	108	109	110	111
	112	113	114	115	116	117	118
	119	120	121	122	123	124	125
	126	127	128	129	130	131	132
	133						
35	36	37	38	39	40	41	42
	43	44	45	46	47	48	49
	50	51	52	53	54	55	56
	57	58	59	60	61	62	63
	64	65	66	67	68	69	70
	71	72	73	74	75	76	77
	78	79	80	81	82	83	84
	85	86	87	88	89	90	91
	92	93	94	95	96	97	98
	99	100	101	102	103	104	105
	106	107	108	109	110	111	112
	113	114	115	116	117	118	119
	120	121	122	123	124	125	126
	127	128	129	130	131	132	133
	134						
36	37	38	39	40	41	42	43
	44	45	46	47	48	49	50
	51	52	53	54	55	56	57
	58	59	60	61	62	63	64
	65	66	67	68	69	70	71
	72	73	74	75	76	77	78
	79	80	81	82	83	84	85
	86	87	88	89	90	91	92
	93	94	95	96	97	98	99
	100	101	102	103	104	105	106

	107	108	109	110	111	112	113
	114	115	116	117	118	119	120
	121	122	123	124	125	126	127
	128	129	130	131	132	133	134
	135						
37	38	39	40	41	42	43	44
	45	46	47	48	49	50	51
	52	53	54	55	56	57	58
	59	60	61	62	63	64	65
	66	67	68	69	70	71	72
	73	74	75	76	77	78	79
	80	81	82	83	84	85	86
	87	88	89	90	91	92	93
	94	95	96	97	98	99	100
	101	102	103	104	105	106	107
	108	109	110	111	112	113	114
	115	116	117	118	119	120	121
	122	123	124	125	126	127	128
	129	130	131	132	133	134	135
	136						
38	39	40	41	42	43	44	45
	46	47	48	49	50	51	52
	53	54	55	56	57	58	59
	60	61	62	63	64	65	66
	67	68	69	70	71	72	73
	74	75	76	77	78	79	80
	81	82	83	84	85	86	87
	88	89	90	91	92	93	94
	95	96	97	98	99	100	101
	102	103	104	105	106	107	108
	109	110	111	112	113	114	115
	116	117	118	119	120	121	122
	123	124	125	126	127	128	129

	130	131	132	133	134	135	136
	137						
39	40	41	42	43	44	45	46
	47	48	49	50	51	52	53
	54	55	56	57	58	59	60
	61	62	63	64	65	66	67
	68	69	70	71	72	73	74
	75	76	77	78	79	80	81
	82	83	84	85	86	87	88
	89	90	91	92	93	94	95
	96	97	98	99	100	101	102
	103	104	105	106	107	108	109
	110	111	112	113	114	115	116
	117	118	119	120	121	122	123
	124	125	126	127	128	129	130
	131	132	133	134	135	136	137
	138						
40	41	42	43	44	45	46	47
	48	49	50	51	52	53	54
	55	56	57	58	59	60	61
	62	63	64	65	66	67	68
	69	70	71	72	73	74	75
	76	77	78	79	80	81	82
	83	84	85	86	87	88	89
	90	91	92	93	94	95	96
	97	98	99	100	101	102	103
	104	105	106	107	108	109	110
	111	112	113	114	115	116	117
	118	119	120	121	122	123	124
	125	126	127	128	129	130	131
	132	133	134	135	136	137	138
	139						
41	42	43	44	45	46	47	48
	49	50	51	52	53	54	55

	56	57	58	59	60	61	62
	63	64	65	66	67	68	69
	70	71	72	73	74	75	76
	77	78	79	80	81	82	83
	84	85	86	87	88	89	90
	91	92	93	94	95	96	97
	98	99	100	101	102	103	104
	105	106	107	108	109	110	111
	112	113	114	115	116	117	118
	119	120	121	122	123	124	125
	126	127	128	129	130	131	132
	133	134	135	136	137	138	139
	140						
42	43	44	45	46	47	48	49
	50	51	52	53	54	55	56
	57	58	59	60	61	62	63
	64	65	66	67	68	69	70
	71	72	73	74	75	76	77
	78	79	80	81	82	83	84
	85	86	87	88	89	90	91
	92	93	94	95	96	97	98
	99	100	101	102	103	104	105
	106	107	108	109	110	111	112
	113	114	115	116	117	118	119
	120	121	122	123	124	125	126
	127	128	129	130	131	132	133
	134	135	136	137	138	139	140
	141						
43	44	45	46	47	48	49	50
	51	52	53	54	55	56	57
	58	59	60	61	62	63	64
	65	66	67	68	69	70	71
	72	73	74	75	76	77	78
	79	80	81	82	83	84	85

	86	87	88	89	90	91	92
	93	94	95	96	97	98	99
	100	101	102	103	104	105	106
	107	108	109	110	111	112	113
	114	115	116	117	118	119	120
	121	122	123	124	125	126	127
	128	129	130	131	132	133	134
	135	136	137	138	139	140	141
	142						
44	45	46	47	48	49	50	51
	52	53	54	55	56	57	58
	59	60	61	62	63	64	65
	66	67	68	69	70	71	72
	73	74	75	76	77	78	79
	80	81	82	83	84	85	86
	87	88	89	90	91	92	93
	94	95	96	97	98	99	100
	101	102	103	104	105	106	107
	108	109	110	111	112	113	114
	115	116	117	118	119	120	121
	122	123	124	125	126	127	128
	129	130	131	132	133	134	135
	136	137	138	139	140	141	142
	143						
45	46	47	48	49	50	51	52
	53	54	55	56	57	58	59
	60	61	62	63	64	65	66
	67	68	69	70	71	72	73
	74	75	76	77	78	79	80
	81	82	83	84	85	86	87
	88	89	90	91	92	93	94
	95	96	97	98	99	100	101
	102	103	104	105	106	107	108
	109	110	111	112	113	114	115

	116	117	118	119	120	121	122
	123	124	125	126	127	128	129
	130	131	132	133	134	135	136
	137	138	139	140	141	142	143
	144						
46	47	48	49	50	51	52	53
	54	55	56	57	58	59	60
	61	62	63	64	65	66	67
	68	69	70	71	72	73	74
	75	76	77	78	79	80	81
	82	83	84	85	86	87	88
	89	90	91	92	93	94	95
	96	97	98	99	100	101	102
	103	104	105	106	107	108	109
	110	111	112	113	114	115	116
	117	118	119	120	121	122	123
	124	125	126	127	128	129	130
	131	132	133	134	135	136	137
	138	139	140	141	142	143	144
	145						
47	48	49	50	51	52	53	54
	55	56	57	58	59	60	61
	62	63	64	65	66	67	68
	69	70	71	72	73	74	75
	76	77	78	79	80	81	82
	83	84	85	86	87	88	89
	90	91	92	93	94	95	96
	97	98	99	100	101	102	103
	104	105	106	107	108	109	110
	111	112	113	114	115	116	117
	118	119	120	121	122	123	124
	125	126	127	128	129	130	131
	132	133	134	135	136	137	138

	139	140	141	142	143	144	145
	146						
48	49	50	51	52	53	54	55
	56	57	58	59	60	61	62
	63	64	65	66	67	68	69
	70	71	72	73	74	75	76
	77	78	79	80	81	82	83
	84	85	86	87	88	89	90
	91	92	93	94	95	96	97
	98	99	100	101	102	103	104
	105	106	107	108	109	110	111
	112	113	114	115	116	117	118
	119	120	121	122	123	124	125
	126	127	128	129	130	131	132
	133	134	135	136	137	138	139
	140	141	142	143	144	145	146
	147						
49	50	51	52	53	54	55	56
	57	58	59	60	61	62	63
	64	65	66	67	68	69	70
	71	72	73	74	75	76	77
	78	79	80	81	82	83	84
	85	86	87	88	89	90	91
	92	93	94	95	96	97	98
	99	100	101	102	103	104	105
	106	107	108	109	110	111	112
	113	114	115	116	117	118	119
	120	121	122	123	124	125	126
	127	128	129	130	131	132	133
	134	135	136	137	138	139	140
	141	142	143	144	145	146	147
	148						
50	51	52	53	54	55	56	57
	58	59	60	61	62	63	64

	65	66	67	68	69	70	71
	72	73	74	75	76	77	78
	79	80	81	82	83	84	85
	86	87	88	89	90	91	92
	93	94	95	96	97	98	99
	100	101	102	103	104	105	106
	107	108	109	110	111	112	113
	114	115	116	117	118	119	120
	121	122	123	124	125	126	127
	128	129	130	131	132	133	134
	135	136	137	138	139	140	141
	142	143	144	145	146	147	148
	149						
51	52	53	54	55	56	57	58
	59	60	61	62	63	64	65
	66	67	68	69	70	71	72
	73	74	75	76	77	78	79
	80	81	82	83	84	85	86
	87	88	89	90	91	92	93
	94	95	96	97	98	99	100
	101	102	103	104	105	106	107
	108	109	110	111	112	113	114
	115	116	117	118	119	120	121
	122	123	124	125	126	127	128
	129	130	131	132	133	134	135
	136	137	138	139	140	141	142
	143	144	145	146	147	148	149
	150						
52	53	54	55	56	57	58	59
	60	61	62	63	64	65	66
	67	68	69	70	71	72	73
	74	75	76	77	78	79	80
	81	82	83	84	85	86	87
	88	89	90	91	92	93	94

	95	96	97	98	99	100	101
	102	103	104	105	106	107	108
	109	110	111	112	113	114	115
	116	117	118	119	120	121	122
	123	124	125	126	127	128	129
	130	131	132	133	134	135	136
	137	138	139	140	141	142	143
	144	145	146	147	148	149	150
	151						
53	54	55	56	57	58	59	60
	61	62	63	64	65	66	67
	68	69	70	71	72	73	74
	75	76	77	78	79	80	81
	82	83	84	85	86	87	88
	89	90	91	92	93	94	95
	96	97	98	99	100	101	102
	103	104	105	106	107	108	109
	110	111	112	113	114	115	116
	117	118	119	120	121	122	123
	124	125	126	127	128	129	130
	131	132	133	134	135	136	137
	138	139	140	141	142	143	144
	145	146	147	148	149	150	151
	152						
54	55	56	57	58	59	60	61
	62	63	64	65	66	67	68
	69	70	71	72	73	74	75
	76	77	78	79	80	81	82
	83	84	85	86	87	88	89
	90	91	92	93	94	95	96
	97	98	99	100	101	102	103
	104	105	106	107	108	109	110
	111	112	113	114	115	116	117
	118	119	120	121	122	123	124

	125	126	127	128	129	130	131
	132	133	134	135	136	137	138
	139	140	141	142	143	144	145
	146	147	148	149	150	151	152
	153						
55	56	57	58	59	60	61	62
	63	64	65	66	67	68	69
	70	71	72	73	74	75	76
	77	78	79	80	81	82	83
	84	85	86	87	88	89	90
	91	92	93	94	95	96	97
	98	99	100	101	102	103	104
	105	106	107	108	109	110	111
	112	113	114	115	116	117	118
	119	120	121	122	123	124	125
	126	127	128	129	130	131	132
	133	134	135	136	137	138	139
	140	141	142	143	144	145	146
	147	148	149	150	151	152	153
	154						
56	57	58	59	60	61	62	63
	64	65	66	67	68	69	70
	71	72	73	74	75	76	77
	78	79	80	81	82	83	84
	85	86	87	88	89	90	91
	92	93	94	95	96	97	98
	99	100	101	102	103	104	105
	106	107	108	109	110	111	112
	113	114	115	116	117	118	119
	120	121	122	123	124	125	126
	127	128	129	130	131	132	133
	134	135	136	137	138	139	140
	141	142	143	144	145	146	147

	148	149	150	151	152	153	154
	155						
57	58	59	60	61	62	63	64
	65	66	67	68	69	70	71
	72	73	74	75	76	77	78
	79	80	81	82	83	84	85
	86	87	88	89	90	91	92
	93	94	95	96	97	98	99
	100	101	102	103	104	105	106
	107	108	109	110	111	112	113
	114	115	116	117	118	119	120
	121	122	123	124	125	126	127
	128	129	130	131	132	133	134
	135	136	137	138	139	140	141
	142	143	144	145	146	147	148
	149	150	151	152	153	154	155
	156						
58	59	60	61	62	63	64	65
	66	67	68	69	70	71	72
	73	74	75	76	77	78	79
	80	81	82	83	84	85	86
	87	88	89	90	91	92	93
	94	95	96	97	98	99	100
	101	102	103	104	105	106	107
	108	109	110	111	112	113	114
	115	116	117	118	119	120	121
	122	123	124	125	126	127	128
	129	130	131	132	133	134	135
	136	137	138	139	140	141	142
	143	144	145	146	147	148	149
	150	151	152	153	154	155	156
	157						
59	60	61	62	63	64	65	66
	67	68	69	70	71	72	73

	74	75	76	77	78	79	80
	81	82	83	84	85	86	87
	88	89	90	91	92	93	94
	95	96	97	98	99	100	101
	102	103	104	105	106	107	108
	109	110	111	112	113	114	115
	116	117	118	119	120	121	122
	123	124	125	126	127	128	129
	130	131	132	133	134	135	136
	137	138	139	140	141	142	143
	144	145	146	147	148	149	150
	151	152	153	154	155	156	157
	158						
60	61	62	63	64	65	66	67
	68	69	70	71	72	73	74
	75	76	77	78	79	80	81
	82	83	84	85	86	87	88
	89	90	91	92	93	94	95
	96	97	98	99	100	101	102
	103	104	105	106	107	108	109
	110	111	112	113	114	115	116
	117	118	119	120	121	122	123
	124	125	126	127	128	129	130
	131	132	133	134	135	136	137
	138	139	140	141	142	143	144
	145	146	147	148	149	150	151
	152	153	154	155	156	157	158
	159						
61	62	63	64	65	66	67	68
	69	70	71	72	73	74	75
	76	77	78	79	80	81	82
	83	84	85	86	87	88	89
	90	91	92	93	94	95	96
	97	98	99	100	101	102	103

	104	105	106	107	108	109	110
	111	112	113	114	115	116	117
	118	119	120	121	122	123	124
	125	126	127	128	129	130	131
	132	133	134	135	136	137	138
	139	140	141	142	143	144	145
	146	147	148	149	150	151	152
	153	154	155	156	157	158	159
	160						
62	63	64	65	66	67	68	69
	70	71	72	73	74	75	76
	77	78	79	80	81	82	83
	84	85	86	87	88	89	90
	91	92	93	94	95	96	97
	98	99	100	101	102	103	104
	105	106	107	108	109	110	111
	112	113	114	115	116	117	118
	119	120	121	122	123	124	125
	126	127	128	129	130	131	132
	133	134	135	136	137	138	139
	140	141	142	143	144	145	146
	147	148	149	150	151	152	153
	154	155	156	157	158	159	160
	161						
63	64	65	66	67	68	69	70
	71	72	73	74	75	76	77
	78	79	80	81	82	83	84
	85	86	87	88	89	90	91
	92	93	94	95	96	97	98
	99	100	101	102	103	104	105
	106	107	108	109	110	111	112
	113	114	115	116	117	118	119
	120	121	122	123	124	125	126
	127	128	129	130	131	132	133

	134	135	136	137	138	139	140
	141	142	143	144	145	146	147
	148	149	150	151	152	153	154
	155	156	157	158	159	160	161
	162						
64	65	66	67	68	69	70	71
	72	73	74	75	76	77	78
	79	80	81	82	83	84	85
	86	87	88	89	90	91	92
	93	94	95	96	97	98	99
	100	101	102	103	104	105	106
	107	108	109	110	111	112	113
	114	115	116	117	118	119	120
	121	122	123	124	125	126	127
	128	129	130	131	132	133	134
	135	136	137	138	139	140	141
	142	143	144	145	146	147	148
	149	150	151	152	153	154	155
	156	157	158	159	160	161	162
	163						
65	66	67	68	69	70	71	72
	73	74	75	76	77	78	79
	80	81	82	83	84	85	86
	87	88	89	90	91	92	93
	94	95	96	97	98	99	100
	101	102	103	104	105	106	107
	108	109	110	111	112	113	114
	115	116	117	118	119	120	121
	122	123	124	125	126	127	128
	129	130	131	132	133	134	135
	136	137	138	139	140	141	142
	143	144	145	146	147	148	149
	150	151	152	153	154	155	156

	157	158	159	160	161	162	163
	164						
66	67	68	69	70	71	72	73
	74	75	76	77	78	79	80
	81	82	83	84	85	86	87
	88	89	90	91	92	93	94
	95	96	97	98	99	100	101
	102	103	104	105	106	107	108
	109	110	111	112	113	114	115
	116	117	118	119	120	121	122
	123	124	125	126	127	128	129
	130	131	132	133	134	135	136
	137	138	139	140	141	142	143
	144	145	146	147	148	149	150
	151	152	153	154	155	156	157
	158	159	160	161	162	163	164
	165						
67	68	69	70	71	72	73	74
	75	76	77	78	79	80	81
	82	83	84	85	86	87	88
	89	90	91	92	93	94	95
	96	97	98	99	100	101	102
	103	104	105	106	107	108	109
	110	111	112	113	114	115	116
	117	118	119	120	121	122	123
	124	125	126	127	128	129	130
	131	132	133	134	135	136	137
	138	139	140	141	142	143	144
	145	146	147	148	149	150	151
	152	153	154	155	156	157	158
	159	160	161	162	163	164	165
	166						
68	69	70	71	72	73	74	75
	76	77	78	79	80	81	82

	83	84	85	86	87	88	89
	90	91	92	93	94	95	96
	97	98	99	100	101	102	103
	104	105	106	107	108	109	110
	111	112	113	114	115	116	117
	118	119	120	121	122	123	124
	125	126	127	128	129	130	131
	132	133	134	135	136	137	138
	139	140	141	142	143	144	145
	146	147	148	149	150	151	152
	153	154	155	156	157	158	159
	160	161	162	163	164	165	166
	167						
69	70	71	72	73	74	75	76
	77	78	79	80	81	82	83
	84	85	86	87	88	89	90
	91	92	93	94	95	96	97
	98	99	100	101	102	103	104
	105	106	107	108	109	110	111
	112	113	114	115	116	117	118
	119	120	121	122	123	124	125
	126	127	128	129	130	131	132
	133	134	135	136	137	138	139
	140	141	142	143	144	145	146
	147	148	149	150	151	152	153
	154	155	156	157	158	159	160
	161	162	163	164	165	166	167
	168						
70	71	72	73	74	75	76	77
	78	79	80	81	82	83	84
	85	86	87	88	89	90	91
	92	93	94	95	96	97	98
	99	100	101	102	103	104	105
	106	107	108	109	110	111	112

	113	114	115	116	117	118	119
	120	121	122	123	124	125	126
	127	128	129	130	131	132	133
	134	135	136	137	138	139	140
	141	142	143	144	145	146	147
	148	149	150	151	152	153	154
	155	156	157	158	159	160	161
	162	163	164	165	166	167	168
	169						
71	72	73	74	75	76	77	78
	79	80	81	82	83	84	85
	86	87	88	89	90	91	92
	93	94	95	96	97	98	99
	100	101	102	103	104	105	106
	107	108	109	110	111	112	113
	114	115	116	117	118	119	120
	121	122	123	124	125	126	127
	128	129	130	131	132	133	134
	135	136	137	138	139	140	141
	142	143	144	145	146	147	148
	149	150	151	152	153	154	155
	156	157	158	159	160	161	162
	163	164	165	166	167	168	169
	170						
72	73	74	75	76	77	78	79
	80	81	82	83	84	85	86
	87	88	89	90	91	92	93
	94	95	96	97	98	99	100
	101	102	103	104	105	106	107
	108	109	110	111	112	113	114
	115	116	117	118	119	120	121
	122	123	124	125	126	127	128
	129	130	131	132	133	134	135
	136	137	138	139	140	141	142

	143	144	145	146	147	148	149
	150	151	152	153	154	155	156
	157	158	159	160	161	162	163
	164	165	166	167	168	169	170
	171						
73	74	75	76	77	78	79	80
	81	82	83	84	85	86	87
	88	89	90	91	92	93	94
	95	96	97	98	99	100	101
	102	103	104	105	106	107	108
	109	110	111	112	113	114	115
	116	117	118	119	120	121	122
	123	124	125	126	127	128	129
	130	131	132	133	134	135	136
	137	138	139	140	141	142	143
	144	145	146	147	148	149	150
	151	152	153	154	155	156	157
	158	159	160	161	162	163	164
	165	166	167	168	169	170	171
	172						
74	75	76	77	78	79	80	81
	82	83	84	85	86	87	88
	89	90	91	92	93	94	95
	96	97	98	99	100	101	102
	103	104	105	106	107	108	109
	110	111	112	113	114	115	116
	117	118	119	120	121	122	123
	124	125	126	127	128	129	130
	131	132	133	134	135	136	137
	138	139	140	141	142	143	144
	145	146	147	148	149	150	151
	152	153	154	155	156	157	158
	159	160	161	162	163	164	165

	166	167	168	169	170	171	172
	173						
75	76	77	78	79	80	81	82
	83	84	85	86	87	88	89
	90	91	92	93	94	95	96
	97	98	99	100	101	102	103
	104	105	106	107	108	109	110
	111	112	113	114	115	116	117
	118	119	120	121	122	123	124
	125	126	127	128	129	130	131
	132	133	134	135	136	137	138
	139	140	141	142	143	144	145
	146	147	148	149	150	151	152
	153	154	155	156	157	158	159
	160	161	162	163	164	165	166
	167	168	169	170	171	172	173
	174						
76	77	78	79	80	81	82	83
	84	85	86	87	88	89	90
	91	92	93	94	95	96	97
	98	99	100	101	102	103	104
	105	106	107	108	109	110	111
	112	113	114	115	116	117	118
	119	120	121	122	123	124	125
	126	127	128	129	130	131	132
	133	134	135	136	137	138	139
	140	141	142	143	144	145	146
	147	148	149	150	151	152	153
	154	155	156	157	158	159	160
	161	162	163	164	165	166	167
	168	169	170	171	172	173	174
	175						
77	78	79	80	81	82	83	84
	85	86	87	88	89	90	91

	92	93	94	95	96	97	98
	99	100	101	102	103	104	105
	106	107	108	109	110	111	112
	113	114	115	116	117	118	119
	120	121	122	123	124	125	126
	127	128	129	130	131	132	133
	134	135	136	137	138	139	140
	141	142	143	144	145	146	147
	148	149	150	151	152	153	154
	155	156	157	158	159	160	161
	162	163	164	165	166	167	168
	169	170	171	172	173	174	175
	176						
78	79	80	81	82	83	84	85
	86	87	88	89	90	91	92
	93	94	95	96	97	98	99
	100	101	102	103	104	105	106
	107	108	109	110	111	112	113
	114	115	116	117	118	119	120
	121	122	123	124	125	126	127
	128	129	130	131	132	133	134
	135	136	137	138	139	140	141
	142	143	144	145	146	147	148
	149	150	151	152	153	154	155
	156	157	158	159	160	161	162
	163	164	165	166	167	168	169
	170	171	172	173	174	175	176
	177						
79	80	81	82	83	84	85	86
	87	88	89	90	91	92	93
	94	95	96	97	98	99	100
	101	102	103	104	105	106	107
	108	109	110	111	112	113	114
	115	116	117	118	119	120	121

	122	123	124	125	126	127	128
	129	130	131	132	133	134	135
	136	137	138	139	140	141	142
	143	144	145	146	147	148	149
	150	151	152	153	154	155	156
	157	158	159	160	161	162	163
	164	165	166	167	168	169	170
	171	172	173	174	175	176	177
	178						
80	81	82	83	84	85	86	87
	88	89	90	91	92	93	94
	95	96	97	98	99	100	101
	102	103	104	105	106	107	108
	109	110	111	112	113	114	115
	116	117	118	119	120	121	122
	123	124	125	126	127	128	129
	130	131	132	133	134	135	136
	137	138	139	140	141	142	143
	144	145	146	147	148	149	150
	151	152	153	154	155	156	157
	158	159	160	161	162	163	164
	165	166	167	168	169	170	171
	172	173	174	175	176	177	178
	179						
81	82	83	84	85	86	87	88
	89	90	91	92	93	94	95
	96	97	98	99	100	101	102
	103	104	105	106	107	108	109
	110	111	112	113	114	115	116
	117	118	119	120	121	122	123
	124	125	126	127	128	129	130
	131	132	133	134	135	136	137
	138	139	140	141	142	143	144
	145	146	147	148	149	150	151

	152	153	154	155	156	157	158
	159	160	161	162	163	164	165
	166	167	168	169	170	171	172
	173	174	175	176	177	178	179
	180						
82	83	84	85	86	87	88	89
	90	91	92	93	94	95	96
	97	98	99	100	101	102	103
	104	105	106	107	108	109	110
	111	112	113	114	115	116	117
	118	119	120	121	122	123	124
	125	126	127	128	129	130	131
	132	133	134	135	136	137	138
	139	140	141	142	143	144	145
	146	147	148	149	150	151	152
	153	154	155	156	157	158	159
	160	161	162	163	164	165	166
	167	168	169	170	171	172	173
	174	175	176	177	178	179	180
	181						
83	84	85	86	87	88	89	90
	91	92	93	94	95	96	97
	98	99	100	101	102	103	104
	105	106	107	108	109	110	111
	112	113	114	115	116	117	118
	119	120	121	122	123	124	125
	126	127	128	129	130	131	132
	133	134	135	136	137	138	139
	140	141	142	143	144	145	146
	147	148	149	150	151	152	153
	154	155	156	157	158	159	160
	161	162	163	164	165	166	167
	168	169	170	171	172	173	174

	175	176	177	178	179	180	181
	182						
84	85	86	87	88	89	90	91
	92	93	94	95	96	97	98
	99	100	101	102	103	104	105
	106	107	108	109	110	111	112
	113	114	115	116	117	118	119
	120	121	122	123	124	125	126
	127	128	129	130	131	132	133
	134	135	136	137	138	139	140
	141	142	143	144	145	146	147
	148	149	150	151	152	153	154
	155	156	157	158	159	160	161
	162	163	164	165	166	167	168
	169	170	171	172	173	174	175
	176	177	178	179	180	181	182
	183						
85	86	87	88	89	90	91	92
	93	94	95	96	97	98	99
	100	101	102	103	104	105	106
	107	108	109	110	111	112	113
	114	115	116	117	118	119	120
	121	122	123	124	125	126	127
	128	129	130	131	132	133	134
	135	136	137	138	139	140	141
	142	143	144	145	146	147	148
	149	150	151	152	153	154	155
	156	157	158	159	160	161	162
	163	164	165	166	167	168	169
	170	171	172	173	174	175	176
	177	178	179	180	181	182	183
	184						
86	87	88	89	90	91	92	93
	94	95	96	97	98	99	100

	101	102	103	104	105	106	107
	108	109	110	111	112	113	114
	115	116	117	118	119	120	121
	122	123	124	125	126	127	128
	129	130	131	132	133	134	135
	136	137	138	139	140	141	142
	143	144	145	146	147	148	149
	150	151	152	153	154	155	156
	157	158	159	160	161	162	163
	164	165	166	167	168	169	170
	171	172	173	174	175	176	177
	178	179	180	181	182	183	184
	185						
87	88	89	90	91	92	93	94
	95	96	97	98	99	100	101
	102	103	104	105	106	107	108
	109	110	111	112	113	114	115
	116	117	118	119	120	121	122
	123	124	125	126	127	128	129
	130	131	132	133	134	135	136
	137	138	139	140	141	142	143
	144	145	146	147	148	149	150
	151	152	153	154	155	156	157
	158	159	160	161	162	163	164
	165	166	167	168	169	170	171
	172	173	174	175	176	177	178
	179	180	181	182	183	184	185
	186						
88	89	90	91	92	93	94	95
	96	97	98	99	100	101	102
	103	104	105	106	107	108	109
	110	111	112	113	114	115	116
	117	118	119	120	121	122	123
	124	125	126	127	128	129	130

	131	132	133	134	135	136	137
	138	139	140	141	142	143	144
	145	146	147	148	149	150	151
	152	153	154	155	156	157	158
	159	160	161	162	163	164	165
	166	167	168	169	170	171	172
	173	174	175	176	177	178	179
	180	181	182	183	184	185	186
	187						
89	90	91	92	93	94	95	96
	97	98	99	100	101	102	103
	104	105	106	107	108	109	110
	111	112	113	114	115	116	117
	118	119	120	121	122	123	124
	125	126	127	128	129	130	131
	132	133	134	135	136	137	138
	139	140	141	142	143	144	145
	146	147	148	149	150	151	152
	153	154	155	156	157	158	159
	160	161	162	163	164	165	166
	167	168	169	170	171	172	173
	174	175	176	177	178	179	180
	181	182	183	184	185	186	187
	188						
90	91	92	93	94	95	96	97
	98	99	100	101	102	103	104
	105	106	107	108	109	110	111
	112	113	114	115	116	117	118
	119	120	121	122	123	124	125
	126	127	128	129	130	131	132
	133	134	135	136	137	138	139
	140	141	142	143	144	145	146
	147	148	149	150	151	152	153
	154	155	156	157	158	159	160

	161	162	163	164	165	166	167
	168	169	170	171	172	173	174
	175	176	177	178	179	180	181
	182	183	184	185	186	187	188
	189						
91	92	93	94	95	96	97	98
	99	100	101	102	103	104	105
	106	107	108	109	110	111	112
	113	114	115	116	117	118	119
	120	121	122	123	124	125	126
	127	128	129	130	131	132	133
	134	135	136	137	138	139	140
	141	142	143	144	145	146	147
	148	149	150	151	152	153	154
	155	156	157	158	159	160	161
	162	163	164	165	166	167	168
	169	170	171	172	173	174	175
	176	177	178	179	180	181	182
	183	184	185	186	187	188	189
	190						
92	93	94	95	96	97	98	99
	100	101	102	103	104	105	106
	107	108	109	110	111	112	113
	114	115	116	117	118	119	120
	121	122	123	124	125	126	127
	128	129	130	131	132	133	134
	135	136	137	138	139	140	141
	142	143	144	145	146	147	148
	149	150	151	152	153	154	155
	156	157	158	159	160	161	162
	163	164	165	166	167	168	169
	170	171	172	173	174	175	176
	177	178	179	180	181	182	183

	184	185	186	187	188	189	190
	191						
93	94	95	96	97	98	99	100
	101	102	103	104	105	106	107
	108	109	110	111	112	113	114
	115	116	117	118	119	120	121
	122	123	124	125	126	127	128
	129	130	131	132	133	134	135
	136	137	138	139	140	141	142
	143	144	145	146	147	148	149
	150	151	152	153	154	155	156
	157	158	159	160	161	162	163
	164	165	166	167	168	169	170
	171	172	173	174	175	176	177
	178	179	180	181	182	183	184
	185	186	187	188	189	190	191
	192						
94	95	96	97	98	99	100	101
	102	103	104	105	106	107	108
	109	110	111	112	113	114	115
	116	117	118	119	120	121	122
	123	124	125	126	127	128	129
	130	131	132	133	134	135	136
	137	138	139	140	141	142	143
	144	145	146	147	148	149	150
	151	152	153	154	155	156	157
	158	159	160	161	162	163	164
	165	166	167	168	169	170	171
	172	173	174	175	176	177	178
	179	180	181	182	183	184	185
	186	187	188	189	190	191	192
	193						
95	96	97	98	99	100	101	102
	103	104	105	106	107	108	109

	110	111	112	113	114	115	116
	117	118	119	120	121	122	123
	124	125	126	127	128	129	130
	131	132	133	134	135	136	137
	138	139	140	141	142	143	144
	145	146	147	148	149	150	151
	152	153	154	155	156	157	158
	159	160	161	162	163	164	165
	166	167	168	169	170	171	172
	173	174	175	176	177	178	179
	180	181	182	183	184	185	186
	187	188	189	190	191	192	193
	194						
96	97	98	99	100	101	102	103
	104	105	106	107	108	109	110
	111	112	113	114	115	116	117
	118	119	120	121	122	123	124
	125	126	127	128	129	130	131
	132	133	134	135	136	137	138
	139	140	141	142	143	144	145
	146	147	148	149	150	151	152
	153	154	155	156	157	158	159
	160	161	162	163	164	165	166
	167	168	169	170	171	172	173
	174	175	176	177	178	179	180
	181	182	183	184	185	186	187
	188	189	190	191	192	193	194
	195						
97	98	99	100	101	102	103	104
	105	106	107	108	109	110	111
	112	113	114	115	116	117	118
	119	120	121	122	123	124	125
	126	127	128	129	130	131	132
	133	134	135	136	137	138	139

	140	141	142	143	144	145	146
	147	148	149	150	151	152	153
	154	155	156	157	158	159	160
	161	162	163	164	165	166	167
	168	169	170	171	172	173	174
	175	176	177	178	179	180	181
	182	183	184	185	186	187	188
	189	190	191	192	193	194	195
	196						
98	99	100	101	102	103	104	105
	106	107	108	109	110	111	112
	113	114	115	116	117	118	119
	120	121	122	123	124	125	126
	127	128	129	130	131	132	133
	134	135	136	137	138	139	140
	141	142	143	144	145	146	147
	148	149	150	151	152	153	154
	155	156	157	158	159	160	161
	162	163	164	165	166	167	168
	169	170	171	172	173	174	175
	176	177	178	179	180	181	182
	183	184	185	186	187	188	189
	190	191	192	193	194	195	196
	197						
99	100	101	102	103	104	105	106
	107	108	109	110	111	112	113
	114	115	116	117	118	119	120
	121	122	123	124	125	126	127
	128	129	130	131	132	133	134
	135	136	137	138	139	140	141
	142	143	144	145	146	147	148
	149	150	151	152	153	154	155
	156	157	158	159	160	161	162
	163	164	165	166	167	168	169

	170	171	172	173	174	175	176
	177	178	179	180	181	182	183
	184	185	186	187	188	189	190
	191	192	193	194	195	196	197
	198						
100	101	102	103	104	105	106	107
	108	109	110	111	112	113	114
	115	116	117	118	119	120	121
	122	123	124	125	126	127	128
	129	130	131	132	133	134	135
	136	137	138	139	140	141	142
	143	144	145	146	147	148	149
	150	151	152	153	154	155	156
	157	158	159	160	161	162	163
	164	165	166	167	168	169	170
	171	172	173	174	175	176	177
	178	179	180	181	182	183	184
	185	186	187	188	189	190	191
	192	193	194	195	196	197	198
	199						

$[S_i] \ 100 \times 1 \quad =$

10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10
	10	10	10	10	10		

$[D_j] \ 1 \times 100 \quad =$

10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10
	10	10	10	10	10		

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Mohammad Rasyid Karomi, lahir di Surabaya pada tanggal 17 Desember 1996. Penulis menempuh pendidikan di SD Al-Hikmah Surabaya (2003-2009), SMP Negeri 3 Peterongan (2009-2012), SMA Negeri 3 Jombang (2012-2014) dan saat ini sedang menempuh pendidikan S1 di Departemen Informatika Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Aktif di berbagai kegiatan di kampus baik akademik maupun non akademik seperti HMTTC, Schematics, UKM Cinta Rebana, hingga Badan Eksekutif

Mahasiswa. Komunikasi dengan penulis dapat melalui email: mrasyidkaromi@gmail.com

